



# **NAVAL POSTGRADUATE SCHOOL**

**MONTEREY, CALIFORNIA**

## **APPLIED CYBER OPERATIONS CAPSTONE PROJECT REPORT**

**IDENTIFYING AND EMBEDDING COMMON  
INDICATORS OF COMPROMISE IN VIRTUAL  
MACHINES FOR LAB-BASED  
INCIDENT RESPONSE EDUCATION**

by

Matthew S. Van Dusen

September 2015

Capstone Advisor:

John D. Fulp

Co-Advisor:

Gurminder Singh

**Approved for public release; distribution is unlimited**

THIS PAGE INTENTIONALLY LEFT BLANK

<b>REPORT DOCUMENTATION PAGE</b>			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.				
<b>1. AGENCY USE ONLY</b> (Leave blank)		<b>2. REPORT DATE</b> September 2015		<b>3. REPORT TYPE AND DATES COVERED</b> Capstone project report
<b>4. TITLE AND SUBTITLE</b> IDENTIFYING AND EMBEDDING COMMON INDICATORS OF COMPROMISE IN VIRTUAL MACHINES FOR LAB-BASED INCIDENT RESPONSE EDUCATION			<b>5. FUNDING NUMBERS</b>	
<b>6. AUTHOR(S)</b> Van Dusen, Matthew S.				
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> Naval Postgraduate School Monterey, CA 93943-5000			<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
<b>9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> N/A			<b>10. SPONSORING / MONITORING AGENCY REPORT NUMBER</b>	
<b>11. SUPPLEMENTARY NOTES</b> The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.				
<b>12a. DISTRIBUTION / AVAILABILITY STATEMENT</b> Approved for public release; distribution is unlimited			<b>12b. DISTRIBUTION CODE</b>	
<b>13. ABSTRACT (maximum 200 words)</b>  Though typical malware delivery vectors, behaviors, and general “attack craft” can be verbally explained and even illustrated, greater familiarity and confidence is imbued in the cyber defender when such theoretical explanations are followed by guided practical exercises that provide realistic scenarios. To demonstrate this, we created seven scenarios utilizing common attack types combined with prominent artifacts for indicators of compromise and prominent incident investigative tools. These scenarios will help facilitate the educational experience for students as well as instill confidence, resulting in more proficient incident response across the field. Should this type of education become a part of the NPS curriculum, additional research can be conducted to reaffirm its true capacity.				
<b>14. SUBJECT TERMS</b> cyber incident response, information technology education			<b>15. NUMBER OF PAGES</b> 83	
			<b>16. PRICE CODE</b>	
<b>17. SECURITY CLASSIFICATION OF REPORT</b> Unclassified	<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b> Unclassified	<b>19. SECURITY CLASSIFICATION OF ABSTRACT</b> Unclassified	<b>20. LIMITATION OF ABSTRACT</b> UU	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)  
Prescribed by ANSI Std. Z39-18

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release; distribution is unlimited**

**IDENTIFYING AND EMBEDDING COMMON INDICATORS OF  
COMPROMISE IN VIRTUAL MACHINES FOR LAB-BASED INCIDENT  
RESPONSE EDUCATION**

Matthew S. Van Dusen

Submitted in partial fulfillment of the  
requirements for the degree of

**MASTER OF SCIENCE IN APPLIED CYBER OPERATIONS**

from the

**NAVAL POSTGRADUATE SCHOOL  
September 2015**

Capstone Project Advisor  
Co-Advisor

John D. Fulp  
Gurminder Singh

Approved by: Cynthia Irvine  
Chair, Cyber Academic Group

THIS PAGE INTENTIONALLY LEFT BLANK

## **ABSTRACT**

Though typical malware delivery vectors, behaviors, and general “attack craft” can be verbally explained and even illustrated, greater familiarity and confidence is imbued in the cyber defender when such theoretical explanations are followed by guided practical exercises that provide realistic scenarios. To demonstrate this, we created seven scenarios utilizing common attack types combined with prominent artifacts for indicators of compromise and prominent incident investigative tools. These scenarios will help facilitate the educational experience for students as well as instill confidence, resulting in more proficient incident response across the field. Should this type of education become a part of the NPS curriculum, additional research can be conducted to reaffirm its true capacity.

THIS PAGE INTENTIONALLY LEFT BLANK



## TABLE OF CONTENTS

<b>I.</b>	<b>INTRODUCTION.....</b>	<b>1</b>
<b>A.</b>	<b>WHAT IS CYBER INCIDENT RESPONSE? .....</b>	<b>1</b>
1.	The Defense Continuum .....	1
2.	Preparation .....	2
3.	Identification .....	3
4.	Containment .....	4
5.	Eradication .....	4
6.	Recovery.....	5
7.	Lessons Learned.....	5
<b>B.</b>	<b>IMPORTANCE OF CYBER TECHNOLOGY IN THE NAVY.....</b>	<b>6</b>
1.	Historical Examples .....	6
2.	Cyberspace Damage Control .....	9
<b>C.</b>	<b>TRAINING DIFFICULTIES.....</b>	<b>10</b>
1.	Labs and Fleet-Wide Scalability.....	10
2.	Threat and Vulnerability Selection .....	10
<b>D.</b>	<b>HOW TO MITIGATE THESE CHALLENGES .....</b>	<b>11</b>
<b>II.</b>	<b>THE MERITS OF VIRTUAL MACHINES FOR INCIDENT RESPONSE EDUCATION .....</b>	<b>13</b>
<b>A.</b>	<b>THE VM CONCEPT .....</b>	<b>13</b>
<b>B.</b>	<b>SELF-CONTAINED AND PORTABLE .....</b>	<b>13</b>
<b>C.</b>	<b>EASILY RECOVERABLE.....</b>	<b>14</b>
<b>D.</b>	<b>INCIDENT RESPONSE TRAINING UTILIZING VMWARE .....</b>	<b>14</b>
<b>III.</b>	<b>PROMINENT ARTIFACTS FOR INDICATORS OF COMPROMISE.....</b>	<b>17</b>
<b>A.</b>	<b>WINDOWS OPERATING SYSTEM .....</b>	<b>17</b>
1.	Registry .....	17
2.	Processes .....	19
3.	Files of Interest.....	19
4.	Network Connections.....	21
5.	Tasks.....	22
6.	Accounts.....	23
7.	Logs .....	24
8.	Advanced Memory and Disk Forensics .....	25
<b>B.</b>	<b>NETWORK SERVICES AND APPLICATIONS .....</b>	<b>27</b>
1.	Dynamic Host Configuration Protocol.....	27
2.	Domain Name System.....	28

3.	Web.....	29
4.	Email .....	30
IV.	PROMINENT INCIDENT INVESTIGATIVE TOOLS.....	33
A.	ABSENCE OF DEEP FORENSICS.....	33
B.	DEDICATED UTILITIES .....	35
1.	Quick Checksum Verifier.....	35
2.	PEView .....	35
3.	Process Explorer .....	36
4.	TCPView .....	36
5.	Regshot.....	36
C.	COMMAND LINE.....	36
1.	Netstat .....	37
2.	Viewing Events .....	39
3.	Viewing Processes and Services .....	40
4.	System File Integrity .....	41
5.	File or Document Integrity.....	42
V.	INCIDENT RESPONSE LAB SCENARIOS.....	45
A.	SCENARIO 1—DOCUMENT AND FILE INTEGRITY.....	45
B.	SCENARIO 2—USE WINDOWS EVENT VIEWER TO LOOK FOR SUSPICIOUS EVENTS .....	47
C.	SCENARIO 3—ANALYSIS OF NETWORK CONNECTIONS .....	49
D.	SCENARIO 4—PROCESS OR SERVICE ANALYSIS.....	51
E.	SCENARIO 5—TASK SCHEDULER ANALYSIS .....	53
F.	SCENARIO 6—EXECUTABLE ANALYSIS.....	55
G.	SCENARIO 7—REGISTRY ANALYSIS .....	57
VI.	CONCLUSIONS AND FUTURE WORK.....	61
A.	CONCLUSIONS .....	61
B.	FUTURE WORK .....	63
	LIST OF REFERENCES.....	65
	INITIAL DISTRIBUTION LIST .....	67

## LIST OF FIGURES

Figure 1.	Netstat Parameters View .....	37
Figure 2.	Netstat View 2.....	38
Figure 3.	Continuation of Netstat -a.....	39
Figure 4.	Command Get-eventlog “Security”-Newest 20.....	40
Figure 5.	Tasklist Command .....	41
Figure 6.	Command SIGVERIF.....	42
Figure 7.	View of Files.....	43
Figure 8.	Command Dir /T:W .....	43
Figure 9.	List of Hashes Taken on All Files Before the Start of the Lab.....	46
Figure 10.	Hash Value for File Lab 2a after a One Paragraph Deletion .....	46
Figure 11.	Security Log.....	48
Figure 12.	Application Log .....	49
Figure 13.	TCPView.....	50
Figure 14.	TCPView—Firefox Connections.....	51
Figure 15.	Process Explorer .....	53
Figure 16.	NortonsUpdate .....	54
Figure 17.	Author’s Username David .....	55
Figure 18.	View of the “Blem” Executable.....	56
Figure 19.	Blem.exe with a TLS Table in the Image Header.....	57
Figure 20.	First Registry Snapshot .....	58
Figure 21.	Second Registry Snapshot.....	59
Figure 22.	MUICache Display .....	59

THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF ACRONYMS AND ABBREVIATIONS

CPU	central processing unit
DDOS	distributed denial of service
DFN-CERT	deutsches forschungsnetz computer emergency response team
DHCP	dynamic host configuration protocol
DLL	dynamic link library
DNS	domain name system
EID	event identification
EPROCESS	executive process
FTK	forensic toolkit
GB	gigabytes
GPS	global positioning system
HTTP	hypertext transfer protocol
IE	Internet Explorer
IMAP	Internet message access protocol
IOC	indicators of compromise
IP	Internet protocol
MD5	message-digest algorithm
NIDS	network intrusion detection system
NKO	Navy knowledge online
NTFS	new technology file system
OS	operating system
OST	offline storage table
PDF	portable document format
PE	portable executable
PFF	personal folder file
PID	process identification
POP	post office protocol
QCV	quick checksum verifier
RAM	random access memory
SCM	service control manager

SSL	secure sockets layer
TCP	transport control protocol
TLS	thread local storage
UDP	user datagram protocol
USB	universal serial bus
VM	virtual machine
VMWare	virtual machine software
WOS	Windows operating system

## **ACKNOWLEDGMENTS**

I would like to thank my wife, Kira, my father, Kevin, my mother, Maureen, and my brother, Dave. It would take another Capstone to list everything you all have done for me. Thank you.

THIS PAGE INTENTIONALLY LEFT BLANK



## **I. INTRODUCTION**

Cyber incident response encompasses all knowledge, skills, abilities and tools that may be employed to detect, analyze and recover from all adverse events that may violate a given security policy. How an organization cultivates its incident response directly correlates with its overall security and defense against potential virtual threats. The practice of utilizing virtual machines (VMs) for the purpose of response training provides useful hands-on experience. It also supports each student's ability to accurately recognize prominent artifacts of compromise.

### **A. WHAT IS CYBER INCIDENT RESPONSE?**

Cyber incident response involves numerous phases and cannot be qualified as one specific event or item. Essentially, cyber incident response is defined as the process of detecting an event that has occurred or is occurring through the resolution of the issue. The process itself and all that it entails is incident response. Ideally, an organization would have durable attack-and-exploit prevention measures in place and thus be resistant to an attack. However, with techniques and motives ever-changing, it is not realistic to think that any organization is immune to threat. Therefore, a plan of action for incident response is necessary.

#### **1. The Defense Continuum**

To say that the process of incident response is cyclical in nature is an understatement. It (ideally) begins with preparation, but it does not end with lessons-learned. It is the response team's responsibility to make the most of those lessons and to apply them toward the preparation (cycle completed) of future incidents. It is a process of constant work to strive for improvement. Continuous effort has to be applied to not only detect and eradicate problems/attacks, but also to remain steps ahead of those who might want to cause harm. The defense continuum essentially includes deterring, preventing, detecting, investigating and recovering from cyber incidents. Because the defense continuum is recurrent, it provides steps linking the process and thus a guide for carrying out the most efficient security possible.

## **2. Preparation**

An excellent place to start when attempting to build an incident response capability is with preparation—the more the better. Although the preparation phase of incident response falls outside the scope of this research, it is valuable to discuss it here for capturing the bigger picture.

Before a team can approach an organization with the intent of minimizing the risk of cyber incidents, they must first ensure their own readiness to respond. Though it may seem relatively simple, or even trite, one of the most important factors of preparation is defining the mission (Luttgens, Mandia, & Pepe, 2014). It is critical to have a well-developed understanding of exactly what the team’s goal is. Defining the “goal” is an important part of scoping what capabilities are required of an incident response team. This scoping then drives the pursuit of tools, skills, training, team membership, organization, procedures, etc., that will yield an effective incident response team.

Having a thorough understanding of an organization’s policies is also useful. Policies are typically documents that establish dicta, or precepts, regarding an organization’s high-level expectations for a range of issues that contribute to accomplishing the organization’s mission. Though personnel issues (i.e., behavior, conflict resolution, grievances) are included among issues addressed by policy, we are most interested in policies pertaining to cyber security. Because incident response services are often provided via a third-party contractor, it is necessary that such external providers understand which policies may inhibit, or even prevent, what otherwise may be the provider’s “standard” response action.

Another significant step during the preparation phase is establishing reliable means of communication. Both internal and external communication must be taken into account. With regard to internal communication, the following options may help ensure the integrity and confidentiality of potentially sensitive discussions and customer data exchanges (Luttgens et al., 2014):

- Encrypt emails
- Properly label all documents and communications

- Monitor conference call participation
- Use case numbers or project names to refer to an investigation.

A team's need for appropriate detection and attack-management tools is irrefutable.

### **3. Identification**

Once an organization is sufficiently prepared for an incident, the next phase is identification. This phase has two responsibilities. The first entails detecting (noticing) “questionable” events. The second entails determining—via appropriate investigation—whether a given cyber-related event (or logically related collection of events) constitutes an incident or not. This is especially significant because, if an actual (i.e., true-positive) incident goes undetected, there will be no subsequent call for any containment or eradication action, resulting in a compromised state of the affected cyber system for an untold amount of time. Therefore, this initial identification of an incident is absolutely critical for the Response Team to make, and to make as early and accurately as possible.

As it pertains to detection in the identification phase, Tondel, Line and Jaatun (2014) reference a study in which incidents were detected in three specific ways:

- By local system and service administrators reporting incidents manually by phone or email
- From automatic security warnings from DFN-CERT or reports from other third party services
- Through local security monitoring mechanisms (Tondel et al., 2014).

Local detection seems to be one of the most significant factors in reports studied. Proximity to the network would allow for more prompt detection and response. In addition, one could make the argument that a thorough examination of the methods and processing of the detection is particularly relevant. This is because a critical look at how identification was made would help the Incident Response Team determine whether there was a false positive reported, thereby conserving invaluable time, energy, and resources.

Luttgens, Pepe, and Mandia (2014) provide a generalized checklist to take into account the main questions the members of an incident response team should ask

themselves upon notification of detection. These questions not only deal with how an incident is detected, but with the detection system itself. Things like whether the detection was done through an automated or manual process; the sources that provided the data; if the source data was validated as accurate; if the source data was preserved; and the detection and error rates all must be taken into consideration during the detection process. What leads and contributes to detection is just as important and relevant as managing that which may come from it. It is worth noting here that this phase will be a main focus for the purpose of this research as the training provided via Virtual Machines will require the user to make such determinations.

#### **4. Containment**

Once an incident has been confirmed to have occurred, the next step is to contain the problem and/or problem area. Initially, the severity of the attack should be assessed in order to effectively determine the degree of the response and resources required (Luttgens et al., 2014).

A key motive during the Containment Phase is to ensure that the attackers are unaware that they have been discovered. There are many reasons for this, but, essentially, the main purpose is to avoid evoking a reaction from the attacker (Luttgens et al., 2014). Should the attackers become aware that they have been discovered, they could alter their behavior and actions in such a way that would make it difficult for investigators to determine the breadth of damage done, thereby also impeding the eventual eradication process.

#### **5. Eradication**

Once the incident has been contained and the Team has been able to establish the degree and reach of the attack, the process of eradication begins.

The purpose of the Eradication Phase is to completely *clean* the affected systems of whatever deleterious incident effects/artifacts have been “deposited” on those systems (Luttgens et al., 2014). Such effects/artifacts are most typically one of either a) added or modified accounts, or b) malware residing in memory and/or storage that remains active

or under future actuation control of a remote controller. Eradication needs to precede recovery as any attempt at recovery without having first eradicated, would be analogous to painting over rust to protect metal from further oxidization (Luttgens et al., 2014). The first step in completing this task is to develop an eradication plan. The main goal of this plan is to remove the attacker's access to, and presence on, the system entirely. It is important to note that this plan should take into consideration the possibility that the attacker may try to regain access during either the Eradication or Recovery Phases (Luttgens et al., 2014).

There are two issues of timing that are essential to this phase: when the eradication is to take place and how long it will last. The former is so that the Team can ensure that every attacker avenue of access (vector) has been thoroughly considered and mitigated/blocked. The latter is important because the longer the phase lasts, the more opportunity the attacker has to reestablish their connection elsewhere in the system (Luttgens et al., 2014).

## **6. Recovery**

The Recovery Phase entails bringing any affected systems back into operation and/or back online. Affected systems should be properly evaluated to ensure that they are free of malicious content, fully functional, secure, and safe for use.

At the end of the Recovery Phase, the entire system should be restored and operate in such a way as to have the appearance that there was never any hindrance to begin with.

## **7. Lessons Learned**

Perhaps one of the most important phases of this cycle is that of Lessons Learned. While proper and detailed documentation is a requirement throughout the execution of every process in this response cycle, it is here especially relevant.

This is the phase in which all incident-related data is marshaled and assessed for potential application going forward. A review of the security controls that were in place at the time the attacker gained access is necessary so as to ascertain what preventive

controls failed, and/or what detection controls might have provided earlier detection. It is important to evaluate both what worked and what failed. This is an opportunity to leverage information gained from the most recent attack response, in order to improve attack prevention and response going forward.

## **B. IMPORTANCE OF CYBER TECHNOLOGY IN THE NAVY**

As the 21st century dawned, nearly every aspect of human activity had become irrevocably intertwined with cyber space, from the public global Internet and its newer military counterparts to GPS precision location, navigation, and timing to financial transactions and personal communications (Wilson, 2014, pp. 8–9).

The advent of cyber (i.e., computer-based) technology has proven to be quite beneficial; it allows for—at its very core—people to complete tasks in seconds that once may have taken days. Due to this convenience and potential for great use, the Navy’s reliance upon technology has paralleled that of its civilian counterpart. However, the potential down-side to this is an entrenched “dependency” that creates a distinct and relatively new kind of vulnerability: a heavily automated system-of-systems that presents a large target surface area to the cyber-savvy enemy. It is this high dependency on systems with numerous vulnerabilities that militates for the Navy to pursue ever-improving risk management capabilities. An excellent way to illustrate this is to examine changes over time and how the Navy has adapted accordingly.

### **1. Historical Examples**

“Computers” first debuted their capability during WWII as establishing their ability to crack codes. “Magic” allowed U.S. forces to decipher Japanese codes in an arena in which Allies had to sometimes interact with Axis parties out of material necessity (Arquilla, 2011). “The age of computers in battle that has unfolded in the past 70 years has proved similar to earlier eras in military history, with these new informational tools pointing to new practices” (Arquilla, 2011, p. 59).

As technology changes, the Navy should be quick and adept at accepting, adapting to, and mastering its ability to operate new technologies in a manner that

maintains or improves its tactical advantage in combat. The cyber environment is just the newest and most fertile new technological change to expand upon in that regard.

Fortunately, it is apparent that the significance of cyber activities has come to the forefront of discussion. The Navy, specifically in recent years, has shown great strides in its overall awareness of the need for skilled technicians in this area. This can be evidenced in the U.S. Navy Information Dominance Roadmap—2013–2028 and the creation of the Task Force Cyber Awakening (TFCA) (Wilson, 2014).

Matthew H. Swartz, director of Communications and Networks Division (N2/N6F1) and TFCA lead spoke to a roundtable in October of 2014 and stated this:

In the last decade, DOD—and specifically the Navy—has been forced to reassess our risk calculus for cyber, to understand from a risk perspective what we need to do to address this growing area. Because of that, we had to make sure we understood the risks of cyber as we move forward (Wilson, 2014, p. 7).

This statement not only reflects the awareness for the need to definitively and actively research the future of the cyber arena, but also the urgency to explore the potential weaknesses within current practice and policy. It also highlights the uncertainty involved in making a conclusive statement regarding what “cyber” actually encompasses.

The fact is that the amount of cyber threats, attacks, and espionage that have transpired within the last decade have pushed the Navy into amending its approach to the cyber environment.

In 2007, there were numerous distributed denial of service (DDoS) attacks was perpetrated against Estonia. Estonia’s government and private-sector information technology infrastructures were put at risk. Russia was first thought to be behind the attacks. This was because the attacks appeared to have come from servers located in Russia. However, because of the anonymity afforded to some hackers, it was later determined that “zombie computers” could have been used and taken control of by anyone. Even Estonians themselves could have been the perpetrators in an attempt to reinforce anti-Russian feelings. No retaliation occurred (Guinchard, 2011).

Canada is another nation that has been struck by cyber attacks. The Treasury Board, the Finance Department and Defense Research and Development Canada were the victims. The impact lasted for approximately two months. The thought at the time was

that the hackers were attempting to gain advance knowledge of the federal budget. Their proposed national budget maintains confidentiality due to the fact that once it is submitted to Parliament, changes cannot be made. This is thought to be the purpose of the attack. Advance knowledge of the federal budget can be used for personal financial gain. Unfortunately, it is practically impossible to officially determine the target of or reason for the attack because whoever perpetrated the attack was ultimately never discovered. Though the attack was traced to China's servers, China may have not been the culprit of the attack. Someone else could have chosen China to be the smoke screen for the attack to help achieve anonymity, or perhaps to misplace blame, or to otherwise confound the incident investigators (Pfleeeger & Pfleeeger, 2012).

There were two main components to the attack perpetrated against Canada. First "executive spear phishing" was used to take over the computers of senior officials. Once this was completed, messages were sent to their internal Information Technology department from individuals posing as the officials. The purpose of this was to gain access of passwords for key systems. "The second component was lacing PDF files with hidden programs and forwarding them to others in the above departments" (Van Dusen, 2013, p. 5). The perpetrators were again relying upon impersonation of the officials who owned the email accounts to convince the recipients of their legitimacy. Upon opening the PDFs, the malicious programs began to execute and simultaneously route confidential and private information back to the attackers (Pfleeeger & Pfleeeger, 2012).

The Stuxnet worm used in 2010 is also an example of the effective use of a cyber attack. This worm was used against nuclear facilities in Iran. Stuxnet had two main purposes. One function was to "force Iran's centrifuges to spin out of control" (Gervais, 2012, p. 37). The other goal for Stuxnet was to simultaneously convince operators that the centrifuges were functioning as though nothing was wrong (Van Dusen, 2013). Stuxnet was also designed to upload information about the system it infected, which effectively made it not only a denial of service cyber weapon, but also a reconnaissance (information gathering) tool for additional attacks (Gervais, 2012). This attack reportedly set back Iran's their nuclear program by a minimum of two years (Slocombe, 2012). Iran chose to retaliate in the summer of 2011. That attack, directed in part against the



Netherlands, was so massive that it “led the Dutch Justice Minister to warn the only secure way to communicate with the Dutch government at that time was with pen, paper, or fax” (Slocombe, 2012, p. 38).

More recently, in 2013, a cyber intrusion was experienced from the United States Army Corps of Engineers National Inventory of Dams. The database that this system encompasses is around 8,100 major dams in the United States. The infiltration of such a system has numerous potential ramifications (Wilson, 2014).

## **2. Cyberspace Damage Control**

The question becomes, now that the Navy has acknowledged the cyber environment as a place for further cultivation, exploration, and investment, what can be done to diminish the likelihood that our cyber vulnerabilities will be exploited? The answer is simple: preparedness. The key to getting ahead of adversaries is to be aware that they are coming, to be aware of the attacks/exploits they are bringing with them, and to be cognizant of the tactics they are likely to.

Cyberspace has been widely accepted as the fifth domain of warfare (1-Sea, 2-Air, 3-Land, 4-Space, and 5-Cyberspace). Eom, Kim, Kim, and Chung (2012) make the argument for the need for cyber superiority as well as for the development of a “cyber warrior.” They contend that the cyber warrior should have thorough knowledge of military policies, cyber strategy, cyber tactics, cyber operations, cyber intelligence collection, as well as cyber attack and cyber defense technology. The concept of having a distinct and separate force dedicated to strictly understanding and operating within the cyber realm is not only interesting, but a practical way of mitigating the risk of attack on our systems while also improving our ability to efficiently attack the systems of our adversaries.

This movement also elevated information technology (IT) from the “nerd” department responsible for keeping an organization’s phones and computers working to the heart of secure data and networking (Wilson, 2014, p. 3).

Not only should all military personnel have a basic comprehension of proper cyber use policies, there should also be a concentration of individuals with a background

in cyber warfare specifically. This is the only way to truly defend against threats while also being capable of perpetrating such threats against our (U.S.) adversaries when it is militarily prudent to do so.

## **C. TRAINING DIFFICULTIES**

### **1. Labs and Fleet-Wide Scalability**

The Naval Postgraduate School (NPS) has a Cyber Battle Lab (CYBL). A cyber battle lab “offers state of the art network and computer systems to build large scale computer networks and computing environments for experimentation” (Terry et al., 2014, p. 1). The CYBL is disconnected from the school’s internal network (intranet) to allow red team operators to attack blue team operators and to practice using a wide variety of tools for offense and defense without the fear of hurting the school’s local area network (NPS, 2015). This type of lab allows students to connect to the virtual environment provided by the cyber battle lab, from anywhere in the world. This type of environment is excellent for teaching the next generation of incident handlers how to detect various types of attacks that can happen on a network, and to mitigate or eradicate such threats as they are discovered. There are only so many workstations that can be logged into the lab at any given time. Though it would be ideal for every network defender in the fleet to have access to a cyber lab, it is just not possible. A cyber battle lab is better suited for “A” schools, “C” schools and universities. Sailors in the fleet need a way to practice using various tools to detect and respond to real world attacks without necessitating access to a high-tech cyber battle lab.

### **2. Threat and Vulnerability Selection**

The list of threats and vulnerabilities that exist in the world of cyber defense is forever changing and growing. This can be overwhelming for many to even think about. The important aspect of cyberattack detection is to know what normal behavior looks like on any given network. It is impossible to know every attack and vulnerability that exists, but practicing with various tools can help an incident responder detect when something is not quite right, and then respond to it. The Navy mostly utilizes Windows operating systems (OS). In order to develop adequate lab scenarios for incident responders to

properly detect and respond to potential incidents, we should look at the areas that are most likely to be attacked, and where an attacker can cause the most damage.

There are numerous different areas that can be attacked on a Windows OS. It is prudent then, given the limited training time available, to focus on high-occurrence and/or high-impact WinOS-based attacks. Particular areas of interest are NTFS and file system analysis, Windows prefetch, event logs, scheduled tasks, the Registry, memory forensics and alternative persistence mechanisms (Luttgens et al., 2014).

#### **D. HOW TO MITIGATE THESE CHALLENGES**

Not all sailors have access to a CYBL. For those who do not, it is important to configure a way for them to receive necessary incident response training. This will ensure that they will be able to perform their duties regarding cyber security efficiently, regardless of circumstance or location. The use of VM is a perfect aid. By creating and providing scenarios directed toward detecting, resolving, and recovering from threats in a VM environment, we are able to provide a hands-on experience to those who may not otherwise have the opportunity to engage safely and constructively with these situations. We have created seven scenarios via VM use that will enable the operator to experience and experiment with different types of threats and how to deal with them.

The benefits of using a VM for educational purposes may not be readily apparent. However, further discussion will show its worth for exactly this purpose. A VM is delivered simplistically; easily reverted to a normative state; and functions without inflicting harm on the physical machine. In addition, use of a VM allows for the instructor to focus the training to items that tend to require more need for investigation. Things like the Registry, processes, tasks, accounts, etc., are all among the items that are most likely to show that an incident has occurred. Network services also possess specific items of interest that should catch a user's attention.

The use of VM also allows the instructor to highlight the best incident investigative tools. This allows the user to become more comfortable with not only what to look for, but what they will need to utilize to identify and remedy a threat or attack. Tools like the QuickChecksum Verifier, PEView, TCP and Regshot may be *common*, yet

are advantageous for the user to be well versed in their use. Having a firm understanding of the command line and all that can be gleaned from it is also beneficial. Because the command line is built into every Windows operating system, it is evident that a user should become quite familiar with it. NETSTAT, viewing events, and system file integrity are all tools that provide a starting point for investigation.

The Navy should consider having an IT incident response team at every command that is trained exclusively to handle cyber incidents. They should be required to conduct incident response training through VM to keep their skills sharp and at the same time, learning how to properly respond to new threats that emerge. The Navy severely lacks adequate training and organization as it pertains to incident response. If cyberspace is the new domain of warfare, we must assume that our networks are already infiltrated. Our Navy's Information System Technicians should and must be more focused on the security of our information systems than just fixing hardware and software issues. This research demonstrates a convenient way to accomplish this training regardless of a sailor's location.

## **II. THE MERITS OF VIRTUAL MACHINES FOR INCIDENT RESPONSE EDUCATION**

The benefits of using virtual machines for the purpose of incident response education are many. Providing students with the opportunity to interact with a threat/attack that has been pre-captured in a virtual machine (VM) gives the student the chance to experiment with the information learned and apply what they have learned in a controlled setting. In addition, the teacher has the ability to view, analyze, and respond to the students' reaction to the variables offered by each scenario.

### **A. THE VM CONCEPT**

Virtualization refers to a concept in which access to a single underlying piece of hardware, like a server, is coordinated so that multiple guest operating systems can share that single piece of hardware, with no guest operating system being aware that it is actually sharing anything at all. A guest operating system appears to the applications running on it as a complete operating system (OS), and the guest OS itself is completely unaware that it's running on top of a layer of virtualization software rather than directly on the physical hardware (Golden, 2008, p. 10).

Essentially, a virtual machine operates on top of the actual machine and can be configured to run a variety of operating systems. There is a plethora of benefits to utilizing such an architecture for the purpose of incident response testing. These are highlighted in the sections that follow.

### **B. SELF-CONTAINED AND PORTABLE**

When computers were first created, their true capacity and potential was likely unforeseen at the time. However, the leaps and bounds technology has made since the first computers' inception are undeniable. The advent of laptops and smart phones meant we were able to complete our computing needs on the go. This characteristic revolutionized how we are able to get things done, as sometimes it is simply not feasible to be physically present at a certain location with so many responsibilities drawing our attentions away.

The same concept is applied to the use of virtual machines. Students will be able to work on and with their attack detection and response skills anywhere, as their virtual machines can be delivered via drop box or flash drive and run on top of VM-compatible laptops. Users can also save, copy, and move their environment from machine to machine. Michael Price (2008) discusses the benefits of copying a VM and files to a flash drive and taking it “with you wherever you go.” Because the educational environment may extend across location—as in the instance when a student is underway—and may not have access to server-based VMs, this is extremely useful. The instructor has the ability to set a return time for when a particular lesson may be handed in. Upon doing so, the instructor then may inspect what has or has not been completed. In addition, a report may accompany the work completed, and what attempts were made can be documented by the student for the instructor’s evaluation.

### **C. EASILY RECOVERABLE**

As is the case when any novice is learning and practicing new skills, there is a chance for something to go wrong. One educational advantage of using virtual machines is that students can gain hands-on experience with actual attacks in a controlled setting. However, mistakes can happen, making it all the more necessary to have precautions in place to ensure that any issue can be readily corrected so work may resume.

One of the best educational outcomes of using Virtual Machines is that they can be promptly reconfigured to a previous—good—state! So, if a student compromises his VM during training, he would be able to *revert or renew* his VM to its previously non-compromised state and continue working. Because the virtual machine and physical host practically exist independently, no damage is suffered by the host OS or physical machine. The VM configuration is *simply* a set of parameters backed up by the Snapshot Function.

### **D. INCIDENT RESPONSE TRAINING UTILIZING VMWARE**

Incident responders throughout the Navy do not all have access to cyber battle labs. There is however, an alternative that can be effective. VMWare software can be downloaded from the VMware website that can install a virtual machine on just about

any workstation (i.e., any host OS). After this software is downloaded, an OS of the users choosing can also be downloaded to run on “top of” the host OS (the guest OS). The Navy could utilize Navy Knowledge Online (NKO) to provide various attacks that can be downloaded onto Virtual Machines from NKO that are placed securely on the website by a trained IT professional. This would allow the command’s incident response plan to be practiced in real time, from the Identification to the Lessons Learned phases. Having this capability would give the network defenders the chance to look at attacks at their leisure, when new tools hit the market, new attacks are placed on NKO, or when tasked to practice/train by their chain of command.

THIS PAGE INTENTIONALLY LEFT BLANK



### **III. PROMINENT ARTIFACTS FOR INDICATORS OF COMPROMISE**

Arguably, one of the most difficult aspects of incident response is the ability to narrow down specific factors of the incident itself. Locating and fixing the problem area(s) is ultimately the point of incident response. Therefore, understanding what the common artifacts are that typically indicate an attack or a place of compromise—as well as where to look for them—is paramount. These items can be considered and either dismissed as irrelevant, or further researched to provide direct answers or suggest leads for further investigation. A superb foundation and place to start can be established by examining the things that are typically good indicators of exploitation.

#### **A. WINDOWS OPERATING SYSTEM**

The Windows Operating System was chosen for the purpose of this project due to its pervasiveness throughout the United States Navy. Because the majority of the computers utilized by the Navy function with this system, it was evident that further analysis and research into its architecture and defense should be the primary focus. Thus, below are enumerated the primary components of the Windows Operating System that are of most utility as indicators of compromise (IOC).

##### **1. Registry**

The Registry is considered central to the Windows Operating System (WOS). It, in and of itself, is the foundation for much of the information available from the system. The Registry is the “primary database of configuration data” and the amount of data that can be gleaned from it is extensive. In addition, as the system itself developed in complexity, so too has the amount of intelligence that can be tracked—and potentially manipulated within the Registry (Luttgens et al., 2014).

The Registry is organized into two different types of hives: system and user-specific. User-specific Registry hives can be found within each user’s profile directory. Hives are generally stored in a single file on a disk. They are not human readable but can be parsed using many different tools. However, one cannot copy hive files with Windows

Explorer. Instead a forensic imaging or some sort of acquisition tool to copy them is required. There are five central Registry hives in the path:

%SYSTEMROOT%\system32\config: SYSTEM, SECURITY, SOFTWARE, SAM, and DEFAULT.

These are hive file names with no extension (Luttgens et al., 2014).

Registry data is stored within a tree formation and is comprised of three items: keys, values, and data (Luttgens et al., 2014). The layout of this information can be viewed within the Windows native Registry editor (regedit.exe). The key represents a path, the value is similar to the naming of a file and the data reflects the true information contained therein. The amount of information that can be monitored, gathered, and maliciously altered within this framework is enormous. There are numerous encodings in which Registry data is stored, and it is because of this that Registry values are human readable whereas other items are not. Per Luttgens et al. (2014), the main rootkit registries are as follows:

- HKEY\_LOCAL\_MACHINE (aka HKLM)
- HKEY\_USERS (aka HKU)
- HKEY\_CURRENT\_USER (aka HKCU)
- HKEY\_CURRENT\_CONFIG (aka HKCC)
- HKEY\_CLASSES\_ROOT.

Registry keys contain a single timestamp. They have an associated LastWriteTime set when the key is created. The timestamp is also updated whenever any values under the keys are changed. However, changes to subkey values do not impact the parent key's timestamp. In addition, Registry timestamps are only affiliated with keys and not values or data therein. Timestamps are helpful in that, by using file system metadata and other evidence that a key was changed during a period of known hacker activity, one could assume the change was made by an unauthorized party.

There are limitations of using a timestamp of a Registry key. If there are a significant number of other keys—particularly within the same or nested paths—updated

within the same second or minute, it is probable that there has been a legitimate change brought about by the operating system, software update, or even a security application such as an antivirus program (Luttgens et al., 2014).

Because the nature of the Registry is that it is the foundation of the Windows Operating System, the amount of evidence that can be attained is vast. However, Luttgens et al. (2014) provide a consolidated list of keys and values they found to be of high significance in their experience with incident response: System Configuration Registry Keys, Shim Cache, Common Auto-Run Registry Keys, and User Hive Registry Keys.

## **2. Processes**

A process is essentially a series of actions taken to reach a desired endpoint/goal. In computing terms, a process refers to an instance of a program that is being executed. In order to accomplish anything on an operating system, a process must be run (Luttgens et al., 2014).

Per Regalado, Harris, and Harper (2015), an important action that needs to transpire regarding processes is that of monitoring. Process Monitor, as its name implies, is a process monitoring tool for Windows that provides a real-time view of the file, system, Registry, and process/thread activity. A useful feature of this monitor is that it allows for the customization of the filter so the user is able to adjust what they are looking at and searching for. This way, the operator is able to focus on the data he/she is most interested in and not be completely overwhelmed with information.

## **3. Files of Interest**

Because there is a plethora of ways in which a hacker may exploit a targeted computer, it is important to be able to have a method for sorting through all of the information available. Files that are inexplicably altered are considered to be “files of interest.” These alterations can be anything from an atypical file name, to an incorrect hash, or an improper location in the file system (Luttgens et al., 2014). In addition, examining first, those files that have historically been specifically targeted is also

preferable from a finding the “signal among the noise” investigative perspective (Luttgens et al., 2014).

An excellent example of this are auto-run keys. These keys are intended to allow for Windows executable files, Dynamic Link Libraries (DLLs), and other components to load upon system boot, user login, and other conditions. Because these keys are executed during the boot and/or user login procedure, and are by design intended to operate transparently, these auto-run keys tend to be a favorite attack target for hackers. Fortunately, there are analytical techniques that can be applied to any type of Registry-based auto-run. Using these types of techniques can aid in narrowing down what data requires further examination.

Jason Luttgens, Kevin Mandia, and Matthew Pepe (2014) discuss the tendency of some incident response personnel to attempt to “eyeball” malicious auto-runs. They advocate against this because they consider it easy to make a mistake via this method. They demonstrate this by listing four keys with a ServiceName, Description and ImagePath for each. They indicate that one of these is malicious and pose the question asking if the reader is able to identify the “bad one.” The three that were legitimate keys had issues that would seemingly be cause for concern. These “anomalies” included spelling errors, lack of a description, the directory from which one is run, and a punctuation error. However, the keys with these attributes were all deemed to be legitimate. The key that was malicious was so because “an attacker had modified the ServiceDll to point to a malicious file, `iprinp32.dll`, rather than the correct DLL file name, `iprip.dll`” (2014).

Their recommendation is to consider the “Registry-based persistence mechanisms” and consider only those that one should include versus exclude. Some of the things to be excluded are persistent binaries signed by publishers and items created outside the—if known—time frame examined. The Team should also consider paths of remaining persistent binaries and critically look at common directories. These are just some ways to focus the initial “triage” examination that is so useful in further funneling-down where additional, more in-depth, incident research should lead (Luttgens et al., 2014).

#### **4. Network Connections**

*The information in this section is entirely based on the work of Luttgens et al. (2014).*

Network monitoring is absolutely crucial for incident detection. Because a machine's connectedness to other machines is via the network, the network provides the main vector of attack/exploit delivery to targeted machines. There are four network data types that can be collected here. They are: alert data, session data, full packet data, and statistical data.

Alert data tend to be quite common for organizations to utilize. Essentially an alert is generated when the alert sensor (e.g., IDS or IPS) detects an event (or several) that has been deemed to be suspicious, if not explicitly malicious. Both network-based and host-based intrusion detection systems can generate these alerts.

Packet header data monitoring is useful but only provides a portion of packet capturing whereas full packet logging is just how it sounds in that it provides a more thorough picture of what is happening. The main purpose of full packet logging is to gather information that has been exchanged between systems so that a transactional "story" can be stitched together, and so that signatures of interest can be identified. It is important to note that at the beginning of an incident investigation, it may be preferable to initially cast a "wider net" and capture and treat all data as potentially important, then become more narrowly selective as the course of the investigation continues.

High-level network statistics offer a "view of what connections, or flows, traverse the network" (Luttgens et al., 2014, p. 188). This data can be most helpful when endpoints involved in an incident are not yet known. It is also useful because it aids in the identification of patterns that may otherwise be missed when looking at individual packet or session data that has not been aggregated into the statistical big picture view.

## 5. Tasks

*The information discussed in this section draws heavily from the work of Luttgens et al. (2014). All quoted material in this section is taken from this work, unless a separate source is explicitly indicated.*

Scheduled tasks provide another avenue by which to research and verify if malicious activity has occurred. “The Windows Task Scheduler provides the ability to automatically execute programs at a specific date and time or on a recurring basis” (Luttgens et al., 2014, p. 305). There are two ways to create scheduled tasks: the “at” command and the “schtasks” command.

There is also a Management Console snap-in for manipulating tasks in the Vista and later version of Windows. However, for the purposes of this project, only *at* and *schtasks* commands will be discussed.

Creating a scheduled task can most easily be done by using the *at* command. This command requires—at the least—local administrator privileges. In general, there are two types of tasks created: those for the local host and those for the target host (remotely created tasks). Tasks created using the *at* command have a relatively simple output. However, there is a fair number of indicators included, such as the dates and times for when certain programs or applications are to run. In addition, tasks can be run in very specified terms. For example, a task can be set to run at 03:00 every Monday, Wednesday, and Saturday.

One important note about creating tasks remotely that needs to be taken into consideration is what time zone the target host is in. “Attackers often run the command `net time \\targetHost` prior to scheduling a remote task, to check the system’s local time” (Luttgens et al., 2014, p. 890).

Whereas the *at* command is a relatively simple one, the *schtasks* command is more “full bodied.” It supports descriptive names, even more complex schedules, and many other features. However, because *schtasks* is a more complicated command, it is used less frequently than the *at* command by attackers. It is important to note that running the *at* command without parameters will only return scheduled tasks already created by

the same command. Running the *schtasks* command in the same fashion will, in contrast, return scheduled tasks created by both the *at* and *schtasks* commands.

Configuration information for scheduled tasks is stored in .job files within the %SYSTEMROOT%\Tasks directory. Using a hex editor one can view the data contained within .job files. Important information can be gleaned by using this method that is human readable. However, a tool that more thoroughly parses the .job files is preferable.

## **6. Accounts**

All computer user accounts come with privileges. These privileges can increase or decrease on two different planes.

When we are attempting to gain access to accounts that have a higher level of privilege than those that we presently have, this is known as vertical privilege escalation. When we are attempting to gain access to different accounts that what we have access to, but are at the same level as the account that we already have access to, this is known as horizontal privilege escalation (Andress & Winterfeld, 2014, p. 4341).

Once an attacker gains access to the system as a user, there are a number of ways in which the vulnerabilities of a system may be exploited. In addition, what may be considered as a standard account on a system may already come equipped with the ability to act as an administrator, thereby immediately providing a hacker with greater access and “mobility” within a system.

According to Luttgens et al. (2014), privilege escalation can be attained through password hash or token dumping followed by password cracking or a variety of other methods. It is relevant to note that a hacker’s priority may be to obtain access to user accounts that may not have administrative authority, but do have access rights to files or resources that will suit the hacker’s needs nonetheless.

Regalado et al. (2015) reference “power permissions” as they pertain to the ability to grant write access, read disposition, and execute disposition. Permissions that allow write access cause “rewriting the service configuration and granting immediate and direct elevation of privilege.” The “read” disposition permissions also have many potential

impacts when being granted to untrusted or semi-trusted users. These include revealing the binary being run, what account is being used to run a service, current state of a service, which services are required, as well as several others. The execute disposition permission has to do with the ability to start, stop, or pause a given service. Suffice it to say, that escalation of privileges is a serious threat to systems (Regalado et al., (2015).

## **7. Logs**

Event logs provide a wealth of information for data mining. By reviewing event logs, one can view such system events as: failed and successful logon attempts, the start and stop of services, alterations to the audit policy, track changes to permissions, and plenty of additional information (Luttgens et al., 2014).

There are three main event logs maintained by Windows: Security, System, and Application. Each of these can be found in a separate file path stored in `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog` (Luttgens et al., 2014). In later versions of Windows, the default paths are:

- Security: `%SYSTEMROOT%\System32\Winevt\Logs\Security.evtx`
- System: `%SYSTEMROOT%\System32\Winevt\Logs\System.evtx`
- Application: `SYSTEMROOT%\System32\Winevt\Logs\Application.evtx`.

According to Luttgens et al. (2014) two essential items required to fully understand event logs are event IDs (EIDs) and tracking and analyzing the events themselves. Every event tracked in Windows has an affiliated ID value. These IDs are useful when trying to research log events. Unfortunately there are hundreds of EIDs that may require additional exploration. However, to compensate for this, Microsoft provides the Events and Errors Message Center, which is a search engine for EID queries. This search engine also provides the ability to search for events containing particular text or which come from specific sources.

Understanding how to read and interpret logon events is also necessary. There are many different things a member of an incident response team may be searching for, like



how a legitimate user accesses his/her system, or how a hacker gained remote access to it. The security event log can provide answers to questions such as these.

Equally important as the Security event log is the System event log. Any time a service is started or stopped, the Service Control Manager (SCM) creates an entry to document the change. Unfortunately, starting, stopping, and changing services occurs frequently throughout the normal cycle of legitimate computer operation, and thus it is difficult to distinguish and weed-out those that are malicious in intent from the many others that otherwise look so similar. When examining a System event log, it is preferable to begin with a “known period of attacker activity” so there will be far fewer log entries to parse and wade through. Another useful tool is to search for “known-bad” service names within the System event log.

The Application event log is the third “OS-native” event log from Windows. This log is especially helpful when logged events in either of the other two logs cannot be easily categorized. Antivirus alerts (in the application log) that are flagged during the time in question can help direct the investigation toward something fruitful, perhaps cueing the investigator where or what specifically to look at in the other two (security, system) logs (Luttgens et al., 2014).

## **8. Advanced Memory and Disk Forensics**

Luttgens et al. (2014) suggest that the contents of memory (RAM) is a proven gold mine of digital evidence. While memory and disk forensics falls outside the scope of this project, it is instructive to address, even if very briefly, this owing to the tremendous amount of information it is likely to contain regarding relevant incident artifacts.

“Memory,” in the case of Windows, entails both Random Access Memory (RAM) as well as hard disk (storage) to run processes. The reason that the hard disk may be involved is in the cases wherein the executable file may be larger than the amount of RAM used to manage its execution. In such cases, most modern OSs (Windows included) maintain swapped-out portions of the executable on the hard drive, swapping these portions (pages) into and out of RAM as calls are made to functions that are needed. In

the Windows OS, the pagefile is the main data structure used to maintain this coordinating information.

“The term ‘physical memory’ simply refers to the contents of RAM” (Luttgens et al., 2014, p. 1027). In order to explore its contents for forensic purposes, one must get an image of it. The size of the image will directly mirror the size of RAM. If there are five gigabytes (GB) of RAM, then the image taken of the physical memory will also be five GB. Most often, software-based tools are required to retrieve memory from a Windows system. Fortunately, most of these tools are portable and can run on a USB drive. In some instances, the Firewire (IEEE1394) port can be used to access the physical memory of a target system, depending on the toolkit used (Luttgens et al., 2014).

Once an image of the physical memory has been attained, it has to be translated into a more simplistic format.

Tools such as the Volatility Framework and Redline can analyze an acquired memory image, recognize the intrinsic structures associated with user-land processes and the kernel, and parse them into human-readable format (Luttgens et al., 2014, p. 1029).

This parsing of information offers the ability to view detailed process listings and all that is associated with these processes (Luttgens et al., 2014).

The pagefile is a secondary storage space that stores memory for processes that cannot fit on the physical memory. It is also connected with RAM in that, as available RAM decreases, the pagefile is more active in moving data back and forth. The default location for the pagefile is %SYSTEMDRIVE%pagefile.sys. However, its location can be moved or even split over various files. If it appears to be missing, the Registry can help. Using the key and value HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management\PagingFiles can help determine if another path has been used. Once it has been located, it cannot simply be copied using Windows Explorer or the command shell. A specific forensic utility is needed to get a copy of the pagefile from a running system.

Crash dumps can be an effective way to analyze what went wrong that caused a system crash. There are three levels of crash dumps: Kernel Memory Dump, Small

Memory Dump, and Complete Memory Dump. The information that can come from these “dumps” can be pivotal during an investigation because it can provide data about the running processes, kernel data as well as other activities that were occurring at the time of the crash.

Reviewing the process listing can also be beneficial to an investigation. Essentially, the process listing provides information as to what is running on a system. Executive Process (EPROCESS) blocks provide kernel data about these running processes. Performing memory forensics can extract things like process ID, process name, process command line, etc., from the list of EPROCESS blocks in kernel memory. It is relevant to note that the tool used will have to map out the parent-child relationships among the data extracted into a hierarchy tree. This is because “processes only track their parent process ID (PID) and not the parent process name or path” (Luttgens et al., 2014, p. 361).

Because deep memory forensics falls outside the scope of this project, further detail is not necessary. However, other memory artifacts that can provide additional evidence are those of network connections, loaded drivers, console command history, strings in memory, and credentials (Luttgens et al., 2014).

## **B. NETWORK SERVICES AND APPLICATIONS**

Though there are many areas within the WOC that may be attacked, there are others that exist outside of this domain. These can be found in network services and applications. Things like the DHCP, DNS, web applications, and Email are all additional surface areas outside of the end device’s OS that present potential for attack.

### **1. Dynamic Host Configuration Protocol**

Dynamic Host Configuration Protocol (DHCP) is a framework which hackers are able to subvert. Essentially, the main purpose of DHCP is to assign Internet Protocol (IP) addresses to devices connected to the network (Luttgens et al., 2014). Therefore, it is no surprise that this mechanism provides an opportunity for malicious activity.

An example of this is DHCP spoofing. It begins with a DHCP client broadcasting a DHCPDISCOVER packet over the network.

All listening and active DHCP servers would respond with DHCPOFFER packets offering a list of configuration parameters. The client then responds to one of the DHCPOFFERs with a DHCPREQUEST packet. The server completes the initialization process by transmitting a DHCPACK packet (Schiffman, O'Donnell, Pennington, & Pollino, 2003, p. 2903).

During the DHCP packet exchange, an IP address, routing and Domain Name System (DNS) information to the DHCP client. Someone with ill intention could impersonate the DHCP server. They would be able to send a valid IP with invalid routing information. This would enable the hacker to view any and all traffic being transmitted over the connection (Schiffman et al., 2003).

## **2. Domain Name System**

The Domain Name System (DNS) is the answer to an ever-broadening and vast Internet. DNS provides a mapping between alpha-numeric names of Internet addresses that are easier to use. This differs from IP addresses, which are purely numeric. An example is [www.metsblog.com](http://www.metsblog.com). Its corresponding IP address is 192.0.79.32. "It is impossible to have one single host file to relate every address with a name and vice versa" (Forouzan, 2010, p. 12206). The resolution to the original host file and its limitations for maintaining such a massive amount of information is the distributed and interactive functionality provided by the DNS. Name-to-IP mapping information is divided into smaller parts and spread across numerous DNS servers located throughout the Internet (Schiffman et al., 2003).

At its very base, the use of the DNS involves query-reply-based communication between name clients and name servers for the purpose of the clients obtaining IP addresses of servers known only by name (Forouzan, 2010). As with any external transmission of data, there is risk of interception and/or alteration of access points. DNS spoofing involves a malicious user "listening" for DNS requests and responding with an IP address of their own choosing (with such choice NOT being a legitimate DNS server).

This would allow the malicious spoofer to “redirect all outgoing traffic through itself before forwarding the traffic along on its correct path; thus creating a “man in the middle” situation. A fully switched network would appear to get in the way of attackers sniffing data not directed to them. However, there are various ways in which a switch can be compromised, resulting in the device being forced to forward or flood packets that would not have otherwise gone to the attacker (Schiffman et al., 2003).

### **3. Web**

Liston and Skoudis (2006) argue that web application attacks have emerged as an ever more popular method of exploiting hosts. Because the Internet is used for things like ecommerce, trading, voting, government, services, etc., it provides a large target of opportunity for nefarious actors. A common misconception is that, if a Secure Sockets Layer (SSL) is being implemented, the connection between the user and a site is impenetrable. While SSL does reinforce authentication and help protect data in transit, it does not mean that there is not any adversarial “work arounds” to get past this security protocol. Account harvesting and the undermining of Web application session tracking can both be done, even with SSL in place.

Account harvesting can be accomplished when a user attempts to log in to a site with the incorrect user identification. It can also occur when a user has the correct user identification but the incorrect password. These are two separate attempt types and may not have been tried by the same perpetrator. This happens because even though both attempts would be denied, the browser header response is different for each. They would look exactly the same with the exception of the end note. The first may return an error 1 notice whereas the latter would return an error 2 notice. It is exactly this variance that an attacker would look for.

Another way in which Web applications can be exploited is via a browser’s cookies. “A cookie is simply an HTTP field that the browser stores on behalf of a Web server” (Liston & Skoudis, 2006, p. 412). There are two types of cookies: persistent and non-persistent. A persistent cookie is one that is written to a local file on the client machine upon the closing of a browser. It will be read by the Web server that created it

when the client returns to that Web server. A non-persistent cookie, on the other hand, is simply “forgotten” (not saved to memory) when the browser is closed.

If a session is tracked via persistent cookies, a nefarious user could edit the local cookie file. According to Skoudis & Liston (2006):

For Internet Explorer, cookies from different servers are stored in their own individual files in the Temporary Internet Files directory for each user...To exploit a session ID based on a persistent cookie, the attacker can log in to the application to get a session ID, close the browser to write the cookie file, edit the cookies using his or her favorite text editor and relaunch the browser, now using the new session ID (Skoudis & Liston, 2006, p. 414)

This can be relatively easily accomplished without the user ever realizing what has occurred.

#### **4. Email**

Email is arguably the most utilized source of commercial communication, and its worldwide usage continues to advance at a rapid pace. It is precisely because of this that it can be used as a prime target for malicious purposes. Email content can be broken down into two sections: body and header. The body consists of content—including attachments—whereas the header is a compilation of sender and recipient email addresses, timestamps, server information and more.

Luttgens, Mandia, and Pepe (2014) reference the four most common types of email they have encountered as Microsoft Outlook for Windows, Microsoft Outlook for OS X, Apple Mail, and Web mail. For the purposes of this project, we will concentrate on Web mail and Microsoft Outlook.

Web mail pertains to services like Gmail, Yahoo!, and AOL. It is not typical for these types of services to store content on a local system. In the case of web mail, most information attained will likely come in the form of browser artifacts. In addition, because Web-based email services are continuously changing, “traditional” tools are less likely to provide a thorough analysis of what may have transpired within an individual account. However, a “well-maintained specialized tool” may offer more of an advantage

when attempting to discover artifacts. Perhaps the most effective method for examining Web mail information is to narrow the time frame of suspected malicious behavior first, and then investigate all aspects of the system that could potentially be impacted. These include browser history, Registry logs, file system, etc. This may provide the best way to get an all-inclusive picture as to what happened.

Microsoft Outlook is a major email client that is utilized by a plethora of companies and agencies. One of the reasons for this is that it supports many different protocols. For example, Outlook can operate with Post Office Protocol (POP), Internet Message Access Protocol (IMAP), Microsoft Exchange, and a number of Web (or HTTP) based services. These are independent of the third party add-ins that provide even more capabilities to Outlook. It is important to note that the directories where Outlook files are located by default can be modified by the user. A good place to begin looking is `HKEY_CURRENT_USER\Software\Microsoft\Office\[version]\Outlook\Search\Catalog`. “Version” here refers to the Office version number.

In addition, Outlook has the capacity for configuring multiple “profiles.” These profiles are stored under a different location than the one aforementioned. A default profile is useful so that it can be eliminated from any additional profiles subsequently created.

The data format that is used by Outlook is the Personal Folder File (PFF). Outlook will also store an email copy offline in the Offline Storage Table (OST). These files can be analyzed by two different types of tools: commercial forensics tools and open source tools. These types of tools have the capability to view the file structure and display the file contents as a tree, as well as compile an executable using “libpff” that can export files (Luttgens et al., 2014).

Just as important as recognizing and making proper use of IOCs is the ability to have a firm grasp of the prominent incident investigative tools used in the field.

THIS PAGE INTENTIONALLY LEFT BLANK



## **IV. PROMINENT INCIDENT INVESTIGATIVE TOOLS**

One of the most essential keys to effective incident response is to be both knowledgeable about, and able to “cultivate” (i.e., modify, improve, extend) the tools involved. For situations where any number of tools can be utilized, the best place to start is by investigating the most prominent tools available. Some items include use of the command line, TCPview, and Regshot, but the list is not necessarily limited to these items. However, as these specific tools pertain to the scenarios described later, these are the tools highlighted and addressed here.

### **A. ABSENCE OF DEEP FORENSICS**

The term digital forensics is an exceptionally broad term and can be used when referring to numerous aspects of the process of incident response. Dezfoli et al. (2013) define digital forensics as the process that “involves collection, preservation, analysis and presentation of evidence from digital sources” (p. 48). Because there are many digital sources, this definition encompasses a wide variety of potential evidence, it is difficult to narrow down all that it entails. It is precisely for this reason that the tools and methods presented here are intended to illustrate a “broad stroke” of all that is available. It is necessary to differentiate between what is to be considered an incident responder versus a digital forensics expert. Essentially, an incident responder is similar to an emergency first responder (medical, rescue, or law enforcement). These are the individuals who know the essentials and can get a person (or organization) back to a stabilized functioning state. Pursuing the medical first responder analogy further, the digital forensics expert is more akin to a surgeon; someone with deeper technical understanding who is able to discern the “root cause” of an ailment and render sophisticated remedies. Incident responders have a certain set of “quick react” skills and tools, whereas a digital forensics expert will rely on more specialized knowledge and tools. Forensic Toolkit (FTK), EnCase, and IDAPro are all examples of more “industrial-strength” malware analysis tools that are often utilized by digital forensic experts. While these tools are slightly beyond the scope of this research, they do require discussion.

FTK is a major digital forensics software tool created by AccessData. It has capabilities that include broad encryption support, comprehensive index and binary search, Internet and chat analysis, and single-node remote investigations. In addition, it also incorporates a standalone disk imager that calculates hash values and can be saved in several format types. This is more of an all-inclusive tool and something that would more likely be used by a digital forensics expert as opposed to an incident responder.

EnCase is another example of a major contender in the market for digital forensics software. It is developed by Guidance Software and offers a great deal in terms of not only thorough investigation but specialized investigation by issue and/or industry. Fields like financial services, healthcare, government and law enforcement can all benefit from this software. In addition, the output of this software can be accepted as evidence in court. Again, this is outside of the focus of this work, but it is worth knowing that this type of product is available.

IDAPro differs from the aforementioned software in that it is more narrowly focused as a multi-processor disassembler and debugger intended to map the execution space of executable code. Per Sikorski and Honig (2012), there are levels of languages used when discussing the operation of malware. These include high-level language, machine code, and low level language (typically assembly). Malware creators typically create their code using a high-level language that is subsequently compiled into machine code in order to be in executable form. This differs from the situation where analysts operate, which usually entails starting at the machine code level, then possibly disassembling it into more human-readable assembly code. Analysts reverse engineer this assembly code to determine how it works as well as its purpose. Reverse engineering refers to breaking down the whole to view and understand its working parts. It is an invaluable tool for an incident response team to utilize.

All of these programs can provide intense and thorough evaluations of a given system or systems. However, the focus of this research has been to demonstrate a bigger picture of the techniques, methodologies, and tools utilized in incident response. The reliance here is on the things that can be learned and understood by the incident responder quickly. It also is to provide a foundation for the argument that the rudimentary

skills of incident response can be learned via interaction with incidents that have been conveniently captured in highly portable VMs.

## **B. DEDICATED UTILITIES**

### **1. Quick Checksum Verifier**

Quick Checksum Verifier (QVC) is a tool that is used to determine if there has been a change in a document or file by using MD-5 or SHA-1 hash functions. A file is loaded into the QVC and a hash value is created. It is important to note that a hash is a numerical value obtained from text and a one-way function/algorithm. The created hash value is saved and compared to future hash values of the file. At any future time one may verify the integrity of the file by re-hashing it and comparing the newly generated hash value to the original for that file. If the values differ, something in the file has changed (lost integrity).

### **2. PEView**

Sikorski and Honig (2012) reference the Portable Executable (PE) view as an excellent tool for viewing the PE file structure. It allows the user to view valuable header information, individual sections, as well as import and export tables. It is through the use of PEView that thread local storage (TLS) callbacks can be effectively utilized.

Many people assume that when they load a program into a debugger, the process will pause as soon as the program executes an instruction. However, this does not always happen. The majority of debuggers actually begin where the PE header defines the program's entry point to be. The benefit of a TLS callback is that it can execute before the designated entry point, and thereby execute secretly.

Basically, TLS allows each thread to maintain a different value for a variable declared using TLS. When TLS is implemented by an executable, the code will typically contain a .tls section in the PE header... TLS supports callback functions for initialization and termination of TLS data objects (Sikorski & Honig, 2012, p. 9310).

A way to discover TLS callbacks is to review the .tls section through PEvent. “Normally functioning” programs do not usually use the .tls section, so if it *is* there it should be considered a red flag.

### **3. Process Explorer**

Another useful tool for researching whether malicious activity is happening is through Process Explorer. Essentially, it is an extremely powerful task manager. Process Explorer monitors and displays processes in a tree format that clearly shows the parent and child relationships among the processes. In addition to the formatting, further organization and clarity of function is displayed with color-coding. New processes—coded in green—tend to be the primary source for initial investigation.

### **4. TCPView**

TCPview is used to detect potential malicious attacks to the network. It has the capability of graphically displaying detailed listings of all endpoints (TCP and UDP) on the system. It can be helpful in the event a process is connecting over a port but what process is making the connection is unknown (Sikorski & Honig, 2012).

### **5. Regshot**

Regshot allows for Registry comparison by taking snapshots of a pre-infected and post-infected system. This allows the incident response team to evaluate what changes are made to the system by the particular malware that was used. Unfortunately, the results will often require plenty of patient scanning and comparison to decipher where the pertinent information is.

## **C. COMMAND LINE**

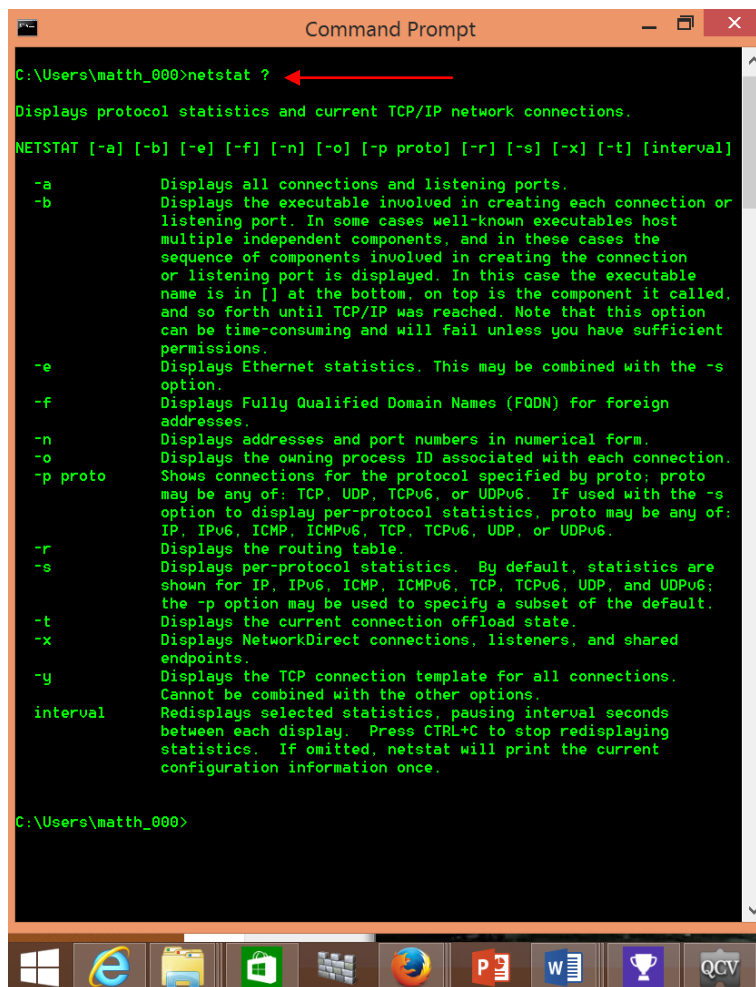
The command line is a great tool for cyber defenders. This tool is built into every Windows operating system. Many prefer the graphical tools because it is easier to look at and decipher information. The fact of the matter is the graphical tools could be compromised or may not be available all together during a cyber emergency. In addition, execution of graphical tools on a potentially infected system may cause new processes

and files to be generated or opened, thereby modifying the operational state of the system. This could actually corrupt forensic data. However, if graphical tools are still preferred, having a knowledge of certain basic commands should be acquired.

## 1. Netstat

Just like the graphical tool TCPview, utilizing this command with its parameters allows an IT professional to monitor network connections. By inputting the command *netstat ?* a person can see all the parameters that netstat provides. This is illustrated in Figures 1, 2 and 3.

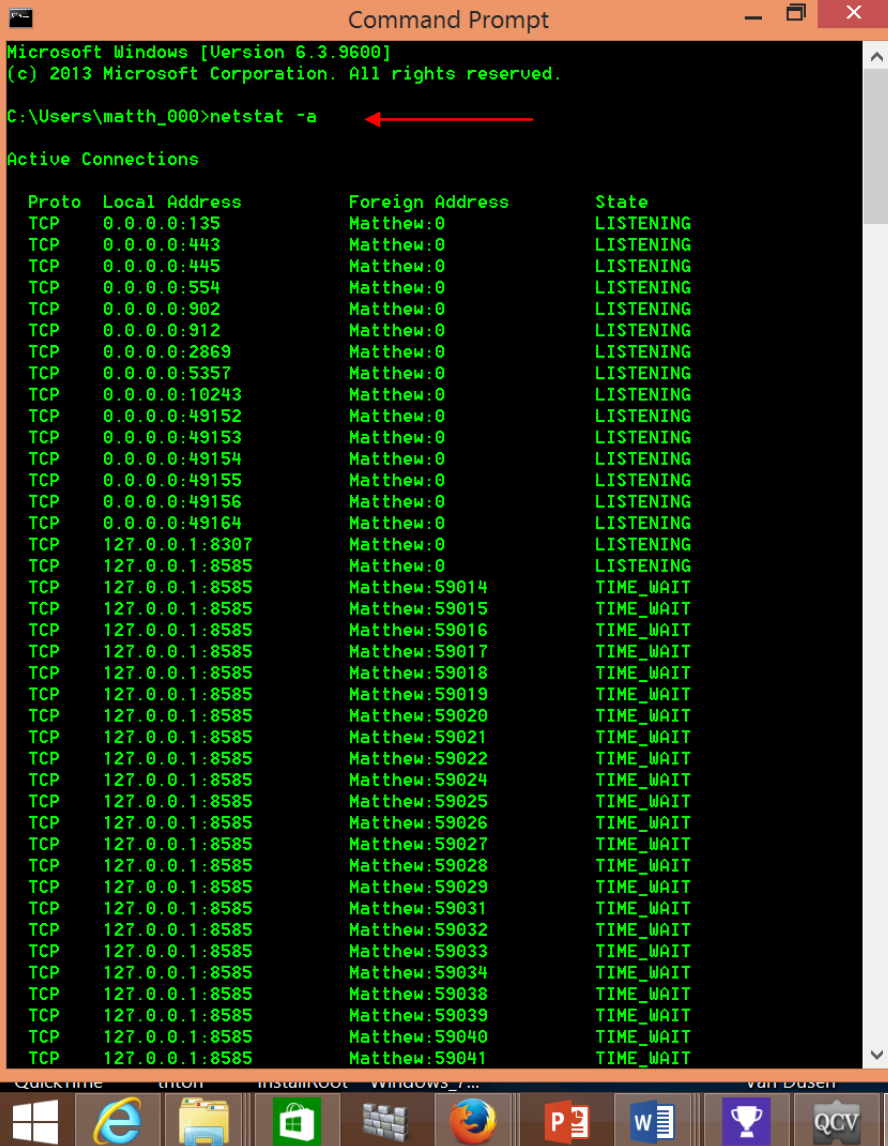
Figure 1. Netstat Parameters View



```
C:\Users\matth_000>netstat ?  
Displays protocol statistics and current TCP/IP network connections.  
NETSTAT [-a] [-b] [-e] [-f] [-n] [-o] [-p proto] [-r] [-s] [-x] [-t] [interval]  
  
-a      Displays all connections and listening ports.  
-b      Displays the executable involved in creating each connection or  
        listening port. In some cases well-known executables host  
        multiple independent components, and in these cases the  
        sequence of components involved in creating the connection  
        or listening port is displayed. In this case the executable  
        name is in [] at the bottom, on top is the component it called,  
        and so forth until TCP/IP was reached. Note that this option  
        can be time-consuming and will fail unless you have sufficient  
        permissions.  
-e      Displays Ethernet statistics. This may be combined with the -s  
        option.  
-f      Displays Fully Qualified Domain Names (FQDN) for foreign  
        addresses.  
-n      Displays addresses and port numbers in numerical form.  
-o      Displays the owning process ID associated with each connection.  
-p proto Shows connections for the protocol specified by proto: proto  
        may be any of: TCP, UDP, TCPv6, or UDPv6. If used with the -s  
        option to display per-protocol statistics, proto may be any of:  
        IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, or UDPv6.  
-r      Displays the routing table.  
-s      Displays per-protocol statistics. By default, statistics are  
        shown for IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, and UDPv6;  
        the -p option may be used to specify a subset of the default.  
-t      Displays the current connection offload state.  
-x      Displays NetworkDirect connections, listeners, and shared  
        endpoints.  
-y      Displays the TCP connection template for all connections.  
        Cannot be combined with the other options.  
interval Redispays selected statistics, pausing interval seconds  
        between each display. Press CTRL+C to stop redisplaying  
        statistics. If omitted, netstat will print the current  
        configuration information once.  
  
C:\Users\matth_000>
```

Netstat? shows parameters used with netstat.

Figure 2. Netstat View 2



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

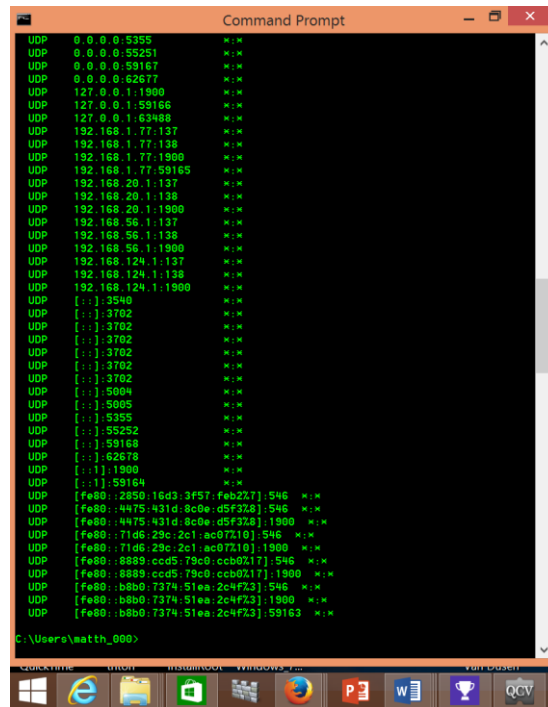
C:\Users\matth_000>netstat -a

Active Connections

Proto Local Address           Foreign Address         State
TCP    0.0.0.0:135               Matthew:0                LISTENING
TCP    0.0.0.0:443               Matthew:0                LISTENING
TCP    0.0.0.0:445               Matthew:0                LISTENING
TCP    0.0.0.0:554               Matthew:0                LISTENING
TCP    0.0.0.0:902               Matthew:0                LISTENING
TCP    0.0.0.0:912               Matthew:0                LISTENING
TCP    0.0.0.0:2869              Matthew:0                LISTENING
TCP    0.0.0.0:5357              Matthew:0                LISTENING
TCP    0.0.0.0:10243             Matthew:0                LISTENING
TCP    0.0.0.0:49152             Matthew:0                LISTENING
TCP    0.0.0.0:49153             Matthew:0                LISTENING
TCP    0.0.0.0:49154             Matthew:0                LISTENING
TCP    0.0.0.0:49155             Matthew:0                LISTENING
TCP    0.0.0.0:49156             Matthew:0                LISTENING
TCP    0.0.0.0:49164             Matthew:0                LISTENING
TCP    127.0.0.1:8307            Matthew:0                LISTENING
TCP    127.0.0.1:8585            Matthew:0                LISTENING
TCP    127.0.0.1:8585            Matthew:59014            TIME_WAIT
TCP    127.0.0.1:8585            Matthew:59015            TIME_WAIT
TCP    127.0.0.1:8585            Matthew:59016            TIME_WAIT
TCP    127.0.0.1:8585            Matthew:59017            TIME_WAIT
TCP    127.0.0.1:8585            Matthew:59018            TIME_WAIT
TCP    127.0.0.1:8585            Matthew:59019            TIME_WAIT
TCP    127.0.0.1:8585            Matthew:59020            TIME_WAIT
TCP    127.0.0.1:8585            Matthew:59021            TIME_WAIT
TCP    127.0.0.1:8585            Matthew:59022            TIME_WAIT
TCP    127.0.0.1:8585            Matthew:59024            TIME_WAIT
TCP    127.0.0.1:8585            Matthew:59025            TIME_WAIT
TCP    127.0.0.1:8585            Matthew:59026            TIME_WAIT
TCP    127.0.0.1:8585            Matthew:59027            TIME_WAIT
TCP    127.0.0.1:8585            Matthew:59028            TIME_WAIT
TCP    127.0.0.1:8585            Matthew:59029            TIME_WAIT
TCP    127.0.0.1:8585            Matthew:59031            TIME_WAIT
TCP    127.0.0.1:8585            Matthew:59032            TIME_WAIT
TCP    127.0.0.1:8585            Matthew:59033            TIME_WAIT
TCP    127.0.0.1:8585            Matthew:59034            TIME_WAIT
TCP    127.0.0.1:8585            Matthew:59038            TIME_WAIT
TCP    127.0.0.1:8585            Matthew:59039            TIME_WAIT
TCP    127.0.0.1:8585            Matthew:59040            TIME_WAIT
TCP    127.0.0.1:8585            Matthew:59041            TIME_WAIT
```

Netstat -a displays all active connections.

Figure 3. Continuation of Netstat -a



```
Command Prompt
C:\Users\math_000>netstat -a

UDP 0.0.0.0:5355 *:*
UDP 0.0.0.0:55251 *:*
UDP 0.0.0.0:59161 *:*
UDP 0.0.0.0:62677 *:*
UDP 127.0.0.1:1900 *:*
UDP 127.0.0.1:59166 *:*
UDP 127.0.0.1:63488 *:*
UDP 192.168.1.77:137 *:*
UDP 192.168.1.77:138 *:*
UDP 192.168.1.77:1900 *:*
UDP 192.168.1.77:59165 *:*
UDP 192.168.20.1:137 *:*
UDP 192.168.20.1:138 *:*
UDP 192.168.20.1:1900 *:*
UDP 192.168.56.1:137 *:*
UDP 192.168.56.1:138 *:*
UDP 192.168.56.1:1900 *:*
UDP 192.168.124.1:137 *:*
UDP 192.168.124.1:138 *:*
UDP 192.168.124.1:1900 *:*
UDP [::]:3540 *:*
UDP [::]:3702 *:*
UDP [::]:3702 *:*
UDP [::]:3702 *:*
UDP [::]:3702 *:*
UDP [::]:3702 *:*
UDP [::]:3702 *:*
UDP [::]:5004 *:*
UDP [::]:5005 *:*
UDP [::]:5355 *:*
UDP [::]:55252 *:*
UDP [::]:59168 *:*
UDP [::]:62678 *:*
UDP [::]:1900 *:*
UDP [::]:59164 *:*
UDP [fe80::2850:16d3:3f57:feb227]:546 *:*
UDP [fe80::4475:431d:8c0e:d5f328]:546 *:*
UDP [fe80::4475:431d:8c0e:d5f328]:1900 *:*
UDP [fe80::71d6:29c:2c1:ac07210]:546 *:*
UDP [fe80::71d6:29c:2c1:ac07210]:1900 *:*
UDP [fe80::8889:ccd5:79cd:ccb0217]:546 *:*
UDP [fe80::8889:ccd5:79cd:ccb0217]:1900 *:*
UDP [fe80::b8b0:7374:51ea:2c4f23]:546 *:*
UDP [fe80::b8b0:7374:51ea:2c4f23]:1900 *:*
UDP [fe80::b8b0:7374:51ea:2c4f23]:59163 *:*
```

## 2. Viewing Events

Windows has what is known as the Event Viewer, which logs many events that are deemed most useful for system monitoring and troubleshooting. It also has an audit setting that allows the operator to tailor what they wish to view. For example: the user may indicate there are specific kinds of events they want logged or they may adjust for the degree of detail provided for the events that are logged. The events that are logged can be viewed through the command line without accessing the tool using powershell. Figure 4 shows one (security) of the three principal logs accessible by Event Viewer. This output is obtainable by use of the command `get-eventlog "security."` The name within the quotations is the name of the log you want to view. This command allows an administrator to view security log entries that have not been deleted. The output shown was further narrowed by appending the command modifier, `"-newest 20,"` to the command so as to only display the most recent 20 log entries.

Figure 4. Command Get-eventlog “Security”-Newest 20

```

PS C:\Windows\system32> get-eventlog "security" -newest 20

```

Index	Time	Entry Type	Source	Instance ID	Message
118980	Aug 25 12:01	SuccessA...	Microsoft-Windows...	4672	Special privileges assigned to new Logon....
118979	Aug 25 12:01	SuccessA...	Microsoft-Windows...	4624	An account was successfully logged on....
118978	Aug 25 12:01	SuccessA...	Microsoft-Windows...	4672	Special privileges assigned to new Logon....
118977	Aug 25 12:01	SuccessA...	Microsoft-Windows...	4624	An account was successfully logged on....
118976	Aug 25 11:59	SuccessA...	Microsoft-Windows...	4672	Special privileges assigned to new Logon....
118975	Aug 25 11:59	SuccessA...	Microsoft-Windows...	4624	An account was successfully logged on....
118974	Aug 25 11:47	SuccessA...	Microsoft-Windows...	4672	Special privileges assigned to new Logon....
118973	Aug 25 11:47	SuccessA...	Microsoft-Windows...	4624	An account was successfully logged on....
118972	Aug 25 11:45	SuccessA...	Microsoft-Windows...	4672	Special privileges assigned to new Logon....
118971	Aug 25 11:45	SuccessA...	Microsoft-Windows...	4624	An account was successfully logged on....
118970	Aug 25 11:45	SuccessA...	Microsoft-Windows...	4672	Special privileges assigned to new Logon....
118969	Aug 25 11:45	SuccessA...	Microsoft-Windows...	4624	An account was successfully logged on....
118968	Aug 25 11:43	SuccessA...	Microsoft-Windows...	4672	Special privileges assigned to new Logon....
118967	Aug 25 11:43	SuccessA...	Microsoft-Windows...	4624	An account was successfully logged on....
118966	Aug 25 11:28	SuccessA...	Microsoft-Windows...	4672	Special privileges assigned to new Logon....
118965	Aug 25 11:28	SuccessA...	Microsoft-Windows...	4624	An account was successfully logged on....
118964	Aug 25 11:28	SuccessA...	Microsoft-Windows...	4672	Special privileges assigned to new Logon....
118963	Aug 25 11:28	SuccessA...	Microsoft-Windows...	4624	An account was successfully logged on....
118962	Aug 25 11:26	SuccessA...	Microsoft-Windows...	4672	Special privileges assigned to new Logon....
118961	Aug 25 11:26	SuccessA...	Microsoft-Windows...	4624	An account was successfully logged on....

Shows the 20 newest entries in the security log

### 3. Viewing Processes and Services

Just like Process Explorer, the tasklist command shows all processes and services running on a machine, as displayed in Figure 5.



Figure 5. Tasklist Command

```
Command Prompt

C:\Users\math_000>tasklist

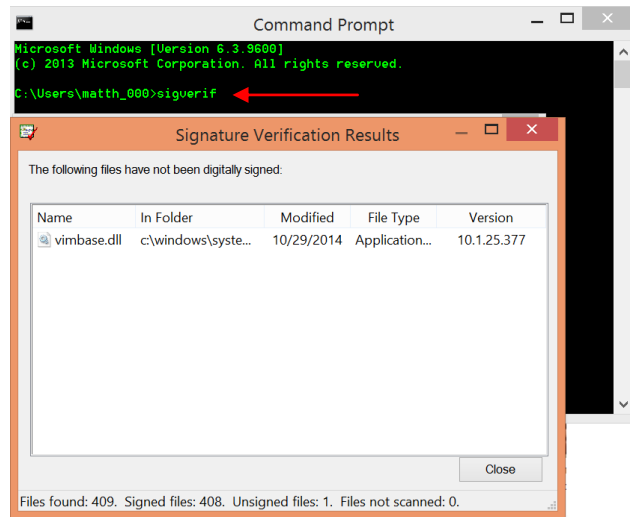
Image Name                   PID Session Name        Session#    Mem Usage
-----
System Idle Process          0 Services             0             4 K
System                        4 Services             0          8,560 K
smss.exe                     332 Services            0          1,056 K
csrss.exe                     504 Services            0          1,056 K
csrss.exe                     576 Console             1          82,424 K
wininit.exe                  584 Services            0          3,832 K
winlogon.exe                 640 Console             1          8,940 K
services.exe                 682 Services            0          8,276 K
lsass.exe                    700 Services            0         18,608 K
svchost.exe                   796 Services            0         14,176 K
svchost.exe                   844 Services            0         12,060 K
dwm.exe                      936 Console             1         144,928 K
svchost.exe                   944 Services            0          33,600 K
svchost.exe                   968 Services            0          64,624 K
svchost.exe                   1016 Services           0          39,412 K
igfxCUIService.exe           616 Services            0          6,224 K
svchost.exe                   760 Services            0         104,460 K
RtkAudioService64.exe        1040 Services            0          5,044 K
RARbg64.exe                  1080 Console             1          10,468 K
RARbg64.exe                  1088 Console             1          10,024 K
WUDFHost.exe                 1096 Services            0          12,012 K
WUDFHost.exe                 1232 Services            0          5,040 K
svchost.exe                  1292 Services            0         21,592 K
wlanext.exe                  1432 Services            0         14,480 K
conhost.exe                  1452 Services            0          2,768 K
epollsv.exe                  1536 Services            0         17,496 K
svchost.exe                   1560 Services            0         21,420 K
svchost.exe                   1592 Services            0         32,620 K
aravac.exe                   1720 Services            0          4,092 K
HESRFS64.exe                 1752 Services            0         2,420 K
officecltcltorun.exe         1776 Services            0         30,092 K
svchost.exe                   1844 Services            0         11,488 K
DptfParticipantProcessors    1864 Services            0          3,268 K
DptfPolicyCriticalService    1884 Services            0          3,228 K
EvtEng.exe                   1912 Services            0         11,620 K
deshout.exe                   1924 Services            0         15,044 K
fntlsrv.exe                   2088 Services            0          3,932 K
ftscamgr.exe                 1832 Services            0         15,220 K
InstallRootService.exe       2224 Services            0         24,136 K
HeciServer.exe               2340 Services            0          5,228 K
RegSvc.exe                   2428 Services            0          6,688 K
schManager.svc               2456 Services            0         14,840 K
svchost.exe                   2532 Services            0         13,880 K
SupportHostAgent.exe          2548 Services            0         49,008 K
```

## All running processes and services

## 4. System File Integrity

It is very common for attackers to manipulate or replace files in order to gain system level access to Windows systems. For this reason, it is a good idea to verify the hash signatures of suspect files. The Windows tool System File Checker verifies the signature of files, and if they are not found or are fraudulent, they will be replaced by the tool. The tool is run with the command `sigverif` as illustrated in Figure 6.

Figure 6. Command SIGVERIF

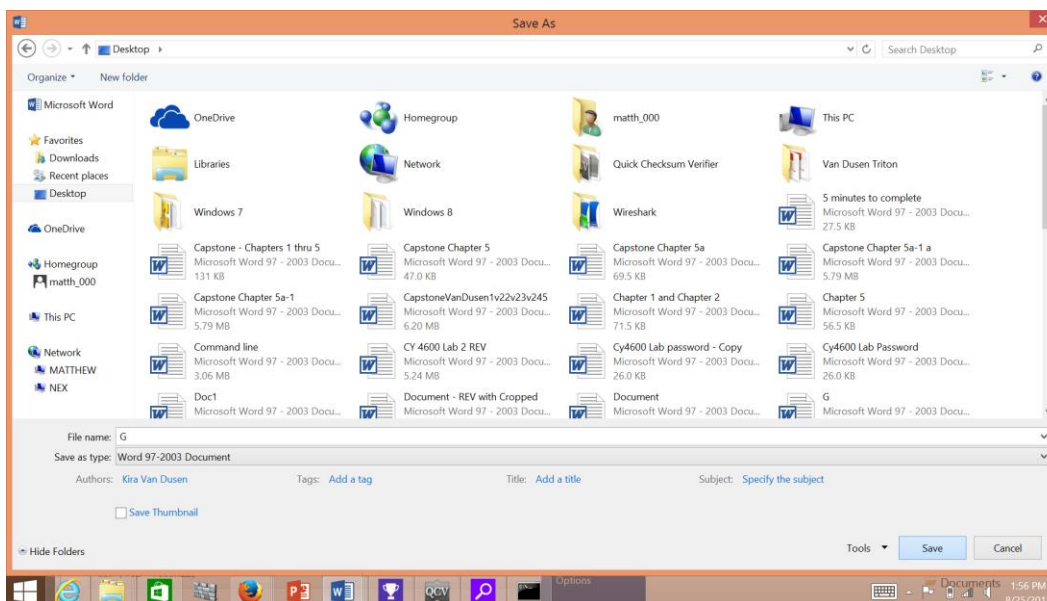


Vimbase.dll has been found to not have a digital signature.

## 5. File or Document Integrity

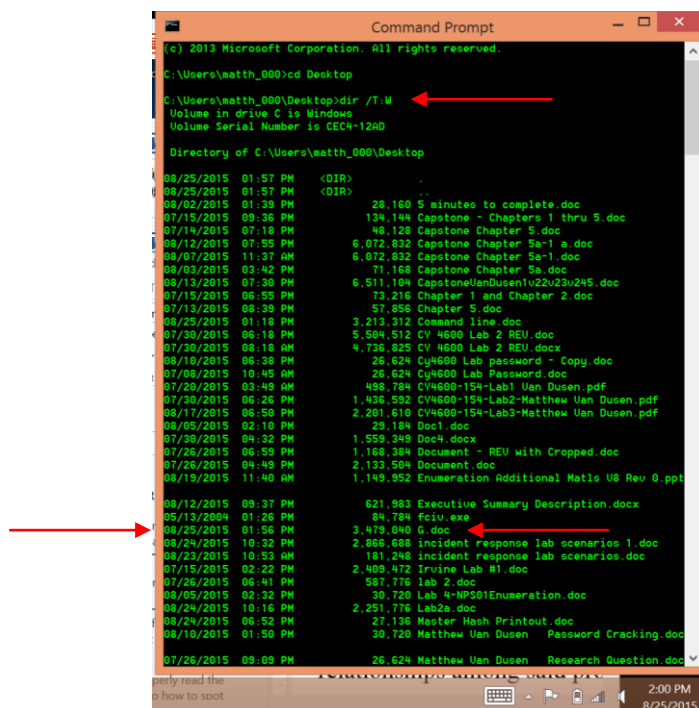
Whereas system files are computer files that the OS uses to function properly, a “regular” file stores information created by users. Much like the Quick Checksum Verifier, the information provided via the `dir` command allows an administrator to tell when files or documents have been modified. If it is known that such files should not have been modified, it would be a good idea to investigate further. This is illustrated by Figures 7 and 8.

Figure 7. View of Files



The document G was modified at 1:56 p.m. on August 25.

Figure 8. Command Dir /T:W



The date and time the document G was last modified.

THIS PAGE INTENTIONALLY LEFT BLANK

## V. INCIDENT RESPONSE LAB SCENARIOS

The scenarios provided in this chapter offer guided practical exercises that provide realistic experiences with compromised systems (captured as VM). These provide the incident responder with the opportunity to make informed decisions regarding which artifacts should be examined as well as which tools are best suited for each situation.

### A. SCENARIO 1—DOCUMENT AND FILE INTEGRITY

- (1) **Target Time to Complete:** 30 minutes
- (2) **Lab Description:** Use the Quick Checksum Verifier (QCV) to confirm the integrity of documents and files.
- (3) **Expected Knowledge, Skills and Abilities to Complete:** The student will need to know what hashes are and that hashes are used by tools such as the QCV to confirm the integrity of files and documents.
- (4) **Source VM and System Requirements to Run:** In order to conduct this lab, VMware Workstation must be installed. This lab scenario uses the Windows 8 operating system.
- (5) **Lab Setup:** The instructor will add 100 files and documents to the desktop and construct a Master Hash Printout for all files and documents. One paragraph of the file Lab2a will be deleted by the instructor. The instructor will then clone the VMware Workstation lab environment and name it Lab 1 for future student use.
- (6) **Lab Write Up and Analysis**
  - (i) **Title:** File and Document Integrity
  - (ii) **Introduction:** The purpose of the lab is to teach the student how to operate a file integrity tool to determine if a file has been changed. The student will need to know how to operate the QCV and how to compare the tool's output hash with the hash on the Master Hash Printout. Also, this lab teaches the student how painful it can be to check the integrity of multiple files with a tool like QCV.
  - (iii) **Tools:** QCV is a tool that is used to determine if there has been a change in a document or file using MD-5 or SHA-1 hash functions. A file is loaded into the QVC and a hash value is created. The created

hash value is saved and compared to future values of the file to determine if the file has been modified.

- (iv) **Results:** Provide screen shots with a brief description that convey pertinent indicators of compromise. This is depicted by Figures 9 and 10.

Figure 9. List of Hashes Taken on All Files Before the Start of the Lab

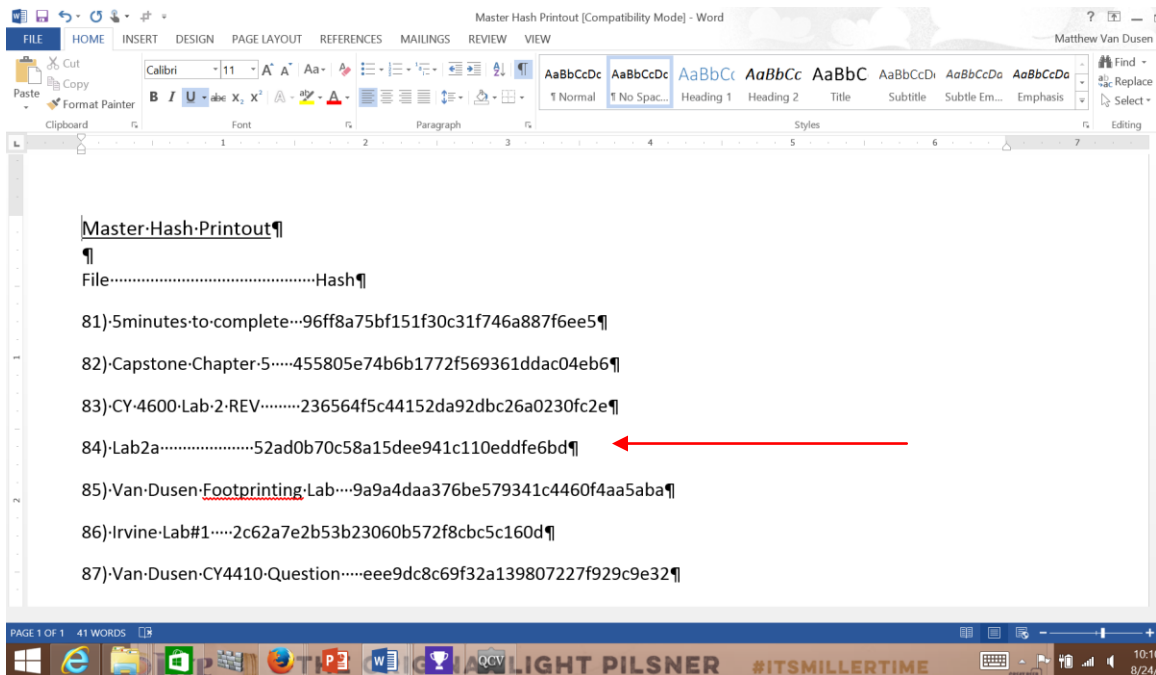
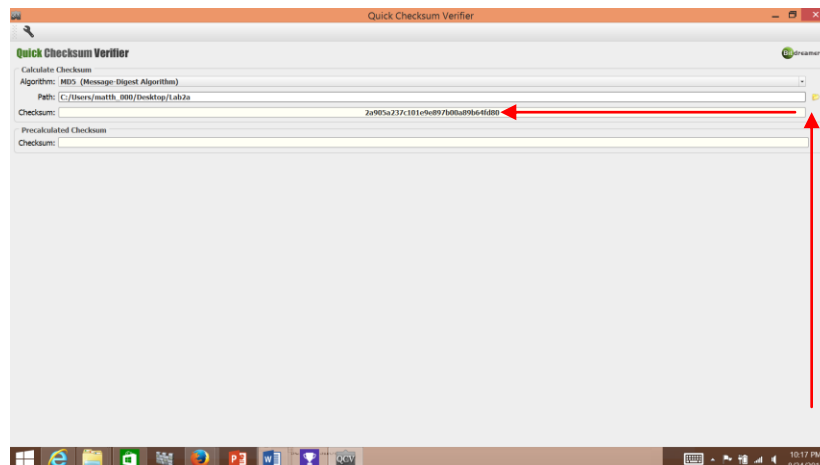


Figure 10. Hash Value for File Lab 2a after a One Paragraph Deletion



To load the file into the tool, click the folder that is to the right of the path window. The student should observe that the hash is different than the hash on the Master Hash File.

**B. SCENARIO 2—USE WINDOWS EVENT VIEWER TO LOOK FOR SUSPICIOUS EVENTS**

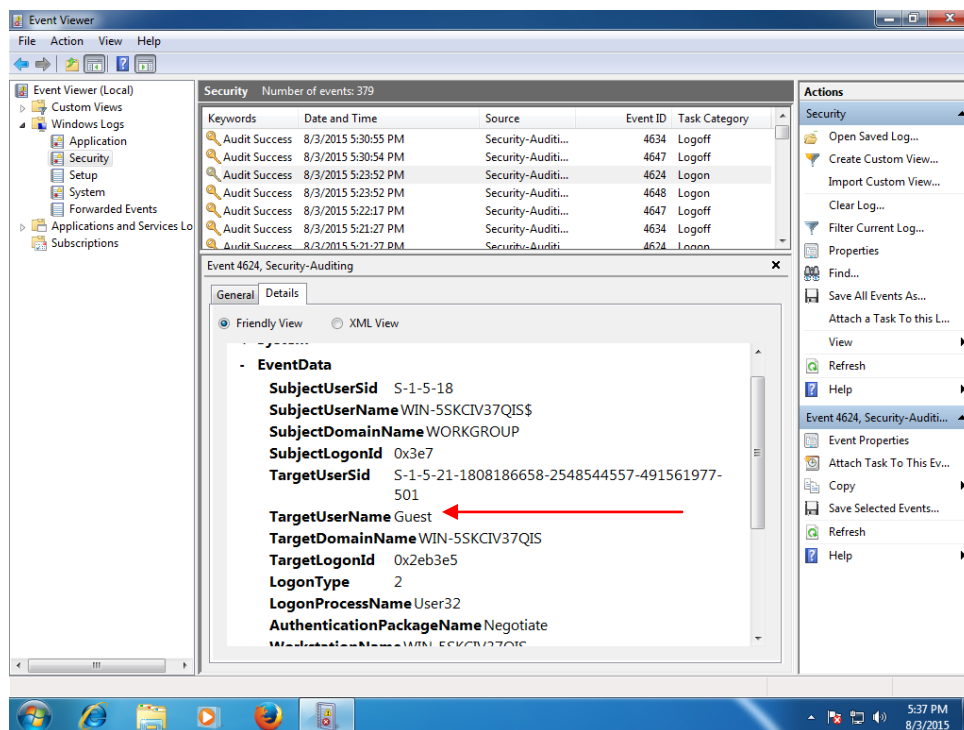
- (1) **Target Time to Complete:** 30 minutes
- (2) **Lab Description:** Utilize Windows Event Viewer to look through 200 Windows events. Use screen shot suspicious events for the lab write up.
- (3) **Expected Knowledge, Skills and Abilities to Complete:** The student will need to know what the application, security, and system event logs are, and that the Event Viewer is the built-in Windows utility used to view/review them. The student will inspect one or more of Windows' three native logs to determine if an incident has occurred.
- (4) **Source VM and System Requirements to Run:** In order to conduct this lab, VMware Horizon Client must be installed. This lab scenario uses a Windows 7 operating system.
- (5) **Lab Set Up:** Malware will be executed in the virtual environment. This malware should be downloaded from the EC-Council CEHV8 Module 07 DVD located in the Viruses and Worms folder. After the instructor confirms that there are 200 events in the event viewer and the malware is executed, a clone of the VMware Workstation environment will be taken by the instructor and named Lab2 for future student use. For this scenario the student will be told there are only four user accounts: Matthew, Kira, George, and Ted. Also, executables are not allowed on this particular machine.
- (6) **Lab Write Up and Analysis**
  - (i) **Title:** Using Event Viewer
  - (ii) **Introduction:** The purpose of the lab is to teach the student how to operate and read the Event Viewer Utility. Looking at logs is a painstaking process and can be very difficult to spot suspicious behavior when you are looking through thousands of line items. It takes a good deal of time to go through 200 events.
  - (iii) **Tools:** Event Viewer was utilized to discover suspicious activity for this lab. Event logs provide a wealth of information for data mining. By reviewing event logs, one can view such system events as: failed

and successful logon attempts, the start and stop of services, alterations to the audit policy, track changes to permissions, and plenty of additional information (Luttgens et al., 2014).

There are three main event logs maintained by Windows: Security, System, and Application. Each of these can be found in a separate file path stored in HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog (Luttgens et al., 2014).

- (iv) **Results:** Provide screen shots with a brief description that convey pertinent indicators of compromise. This can be seen in Figures 11 and 12.

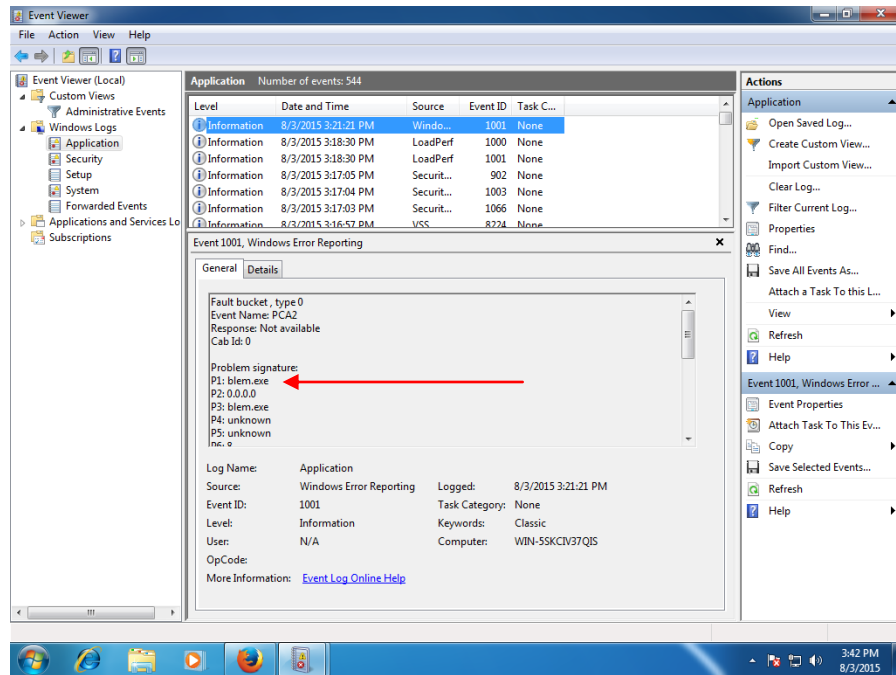
Figure 11. Security Log



This username is not one of the four authorized accounts for this machine.



Figure 12. Application Log



Here it shows the problem signature blem.exe. Executables are unauthorized.

### C. SCENARIO 3—ANALYSIS OF NETWORK CONNECTIONS

- (1) **Target Time to Complete:** 30 minutes
- (2) **Lab Description:** Use TCPview to look through two weeks of saved network connections. Screen shot suspicious connections for the lab write up.
- (3) **Expected Knowledge, Skills and Abilities to Complete:** The student must know what the TCPView utility is used for monitoring network connections specifically TCP and UDP endpoints.
- (4) **Source VM and System Requirements to Run:** In order to conduct this lab VMware Workstation must be installed. This lab scenario uses the Windows 7 operating system.
- (5) **Lab Setup:** Malware will be executed in the virtual environment. The malware is downloaded from the EC-Council CEHV8 Module 07 DVD located in the Viruses and Worms folder. Once the instructor has sufficient saved network connections in the TCPview tool and the malware has been executed the instructor will clone the VMware Workstation lab environment and name it Lab3 for future student use.

## Lab Write Up and Analysis:

- (i) **Title:** Analysis of Network Connections Introduction
- (ii) **Introduction:** The purpose of the lab is to teach the student how to operate and decipher network connections using TCPView. Sometimes it can be very difficult to determine if a network connection is malicious or not because many attackers are good at hiding malicious connections in plain sight. In order to detect malicious network connections defenders must constantly monitor network activity with tools.
- (iii) **Tools:** TCPview is used to detect potential malicious attacks to the network. It has the capability of graphically displaying detailed listings of all endpoints (TCP and UDP) on the system. It can be helpful in the event that a process connects over a port but what process is making the connection is unknown (Sikorski. & Honig, 2012)
- (iv) **Results:** Provide screen shots with a brief description that convey pertinent indicators of compromise. These can be viewed in Figures 13 and 14.

Figure 13. TCPView

Process	PID	Protocol	Local Address	Local Port	Remote Address	Remote Port	State	Sent Packets	Sent Bytes	Received Bytes
24474964.DXE	6322	TCP	WIN-5SKCIV37QIS	49234	mx1.hotmail.com	smtp	SYN_SENT			
lsass.exe	508	TCP	WIN-5SKCIV37QIS	49156	WIN-5SKCIV37QIS	0	LISTENING			
lsass.exe	508	TCPV6	WIN-5SKCIV37QIS	49156	WIN-5SKCIV37QIS	0	LISTENING			
services.exe	500	TCP	WIN-5SKCIV37QIS	49155	WIN-5SKCIV37QIS	0	LISTENING			
services.exe	500	TCPV6	WIN-5SKCIV37QIS	49155	WIN-5SKCIV37QIS	0	LISTENING			
svchost.exe	724	TCP	WIN-5SKCIV37QIS	epmap	WIN-5SKCIV37QIS	0	LISTENING			
svchost.exe	768	TCP	WIN-5SKCIV37QIS	49153	WIN-5SKCIV37QIS	0	LISTENING			
svchost.exe	924	TCP	WIN-5SKCIV37QIS	49154	WIN-5SKCIV37QIS	0	LISTENING			
svchost.exe	768	UDP	WIN-5SKCIV37QIS	bootpc	*	*				
svchost.exe	1080	UDP	WIN-5SKCIV37QIS	rtp	*	*				
svchost.exe	1224	UDP	WIN-5SKCIV37QIS	llnr	*	*				
svchost.exe	724	TCPV6	WIN-5SKCIV37QIS	epmap	WIN-5SKCIV37QIS	0	LISTENING			
svchost.exe	768	TCPV6	WIN-5SKCIV37QIS	49153	WIN-5SKCIV37QIS	0	LISTENING			
svchost.exe	924	TCPV6	WIN-5SKCIV37QIS	49154	WIN-5SKCIV37QIS	0	LISTENING			
svchost.exe	1080	UDPV6	WIN-5SKCIV37QIS	123	*	*				
svchost.exe	768	UDPV6	WIN-5SKCIV37QIS	546	*	*				
svchost.exe	1224	UDPV6	WIN-5SKCIV37QIS	5355	*	*				
System	4	TCP	WIN-5SKCIV37QIS	netbios-ssn	WIN-5SKCIV37QIS	0	LISTENING			
System	4	TCP	WIN-5SKCIV37QIS	microsoft-ds	WIN-5SKCIV37QIS	0	LISTENING			
System	4	UDP	WIN-5SKCIV37QIS	netbios-ns	*	*		9	612	
System	4	UDP	WIN-5SKCIV37QIS	netbios-dgm	*	*				
System	4	TCPV6	WIN-5SKCIV37QIS	microsoft-ds	WIN-5SKCIV37QIS	0	LISTENING			
wininit.exe	412	TCP	WIN-5SKCIV37QIS	49152	WIN-5SKCIV37QIS	0	LISTENING			
wininit.exe	412	TCPV6	WIN-5SKCIV37QIS	49152	WIN-5SKCIV37QIS	0	LISTENING			

Endpoints: 25   Established: 0   Listening: 15   Time Wait: 0   Close Wait: 0

7:05 PM 8/3/2015

The top two process names do not make sense.

Figure 14. TCPView—Firefox Connections

Process	PID	Protocol	Local Address	Local Port	Remote Address	Remote Port	State	Sent Packets	Sent Bytes
firefox.exe	2508	TCP	win-5skov37qis.lo...	49191	5.10.84.137-static...	http	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49192	5.10.84.137-static...	http	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49193	a23-72-180-19.de...	http	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49194	a23-72-180-19.de...	http	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49195	a23-72-180-19.de...	http	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49196	a23-72-180-19.de...	http	ESTABLISHED	1	1,652
firefox.exe	2508	TCP	win-5skov37qis.lo...	49197	a23-72-180-19.de...	http	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49198	a23-72-180-19.de...	http	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49201	a23-72-180-19.de...	http	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49202	mc.yandex.ru	https	ESTABLISHED	1	1,401
firefox.exe	2508	TCP	win-5skov37qis.lo...	49203	ruq04c29-in14.1...	https	ESTABLISHED	2	447
firefox.exe	2508	TCP	win-5skov37qis.lo...	49204	5.10.84.137-static...	http	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49205	5.10.84.137-static...	http	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49206	5.10.84.137-static...	http	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49207	5.10.84.137-static...	http	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49209	ruq04c19-in18.1e...	https	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49211	ruq0502-in130.1...	https	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49212	ruq04c29-in13.1e...	https	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49213	ruq0502-in125.1...	https	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49214	ruq04c19-in128.1...	https	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49215	ruq0502-in131.1...	https	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49217	ruq0501-in14.1e...	http	ESTABLISHED	1	31
firefox.exe	2508	TCP	win-5skov37qis.lo...	49219	173.194.8.236	https	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49220	74.125.224.83	https	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49221	74.125.224.83	http	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49223	74.125.239.129	https	ESTABLISHED	10	2,768
firefox.exe	2508	TCP	win-5skov37qis.lo...	49224	74.125.239.39	http	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49226	74.125.28.156	https	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49227	ruq0502-in131.1...	https	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49229	63.245.217.43	https	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49230	23.75.41.32	https	ESTABLISHED		
firefox.exe	2508	TCP	win-5skov37qis.lo...	49231	23.72.180.96	http	ESTABLISHED		
lsass.exe	508	TCP	WIN-5SKOV37QIS	49156	WIN-5SKOV37QIS	0	LISTENING		
lsass.exe	508	TCPV6	[0.0.0.0:0.0.0.0]	49156	[0.0.0.0:0.0.0.0]	0	LISTENING		
services.exe	500	TCP	WIN-5SKOV37QIS	49155	WIN-5SKOV37QIS	0	LISTENING		
services.exe	500	TCPV6	[0.0.0.0:0.0.0.0]	49155	[0.0.0.0:0.0.0.0]	0	LISTENING		
svchost.exe	724	TCP	WIN-5SKOV37QIS	epnmap	WIN-5SKOV37QIS	0	LISTENING		
svchost.exe	768	TCP	WIN-5SKOV37QIS	49153	WIN-5SKOV37QIS	0	LISTENING		
svchost.exe	924	TCP	WIN-5SKOV37QIS	49154	WIN-5SKOV37QIS	0	LISTENING		
svchost.exe	768	UDP	WIN-5SKOV37QIS	bootpc	*	*			
svchost.exe	1080	UDP	WIN-5SKOV37QIS	ftp	*	*			

These Firefox connections are suspicious. The firefox.exe processes have the same process identifier and are communicating on local ports 49191-4923. Forty TCP endpoints is very high for one process identifier. The remote ports are http or https, which are very common ports used by attackers. These remote ports are usually not being watched or blocked by defenders due to their heavy network traffic. In addition, these ports are usually open for authorized use.

#### D. SCENARIO 4—PROCESS OR SERVICE ANALYSIS

- (1) **Target Time to Complete:** 30 minutes
- (2) **Lab Description:** Monitor processes and services with Process Explorer. Suspicious events are captured using screen shots for the lab write up.
- (3) **Expected Knowledge, Skills and Abilities to Complete:** The student will need to know how to properly read the process explorer output and also how to spot service or processes that may be malicious. The student will need to know what the colors pink, blue, green, and red mean when the process viewer highlights the various services and processes. Also, the

student will need to know what a malicious service or process looks like within the process viewer.

- (4) **Source VM and System Requirements to Run:** In order to conduct this lab VMware Workstation must be installed. This lab scenario is using a Windows 7 operating system
- (5) **Lab Setup:** Malware will be executed in the virtual environment. The malware is downloaded from the EC-Council CEHV8 Module 07 DVD located in the Viruses and Worms folder. Processes and services from the two last weeks of the computer's operations will be saved inside process explorer. Once the instructor has sufficient processes and services in the tool and the malware has been executed the instructor will clone the VMware Workstation lab environment and name it Lab 4 for future student use.
- (6) **Lab Write Up and Analysis:**
  - (i) **Title:** Process and Service Monitoring
  - (ii) **Introduction:** The purpose of the lab is to teach the student how to operate and decipher processes and services using process explorer. Sometimes it can be very difficult to determine if a service or process connection is malicious or not, because many attackers are good at hiding malicious processes and services in plain sight. In order to detect malicious services and processes network defenders must constantly monitor them with various tools.
  - (iii) **Tools:** Process Explorer monitors and displays processes in a tree format that clearly shows the parent and child relationships among the said processes. In addition to the formatting, further organization and clarity of function is provided through the use of color-coded displays. New processes—coded in green—tend to be the primary source for initial investigation (Sikorski & Honig, 2012). This is all depicted in Figure 15.
  - (iv) **Results:** Provide screen shots with a brief description that convey pertinent indicators of compromise.

Figure 15. Process Explorer

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
csrss.exe	0.01	1,260 K	2,740 K	360		
wininit.exe		868 K	2,576 K	412		
services.exe		3,644 K	5,692 K	500		
svchost.exe		2,508 K	5,400 K	644	Host Process for Windows S...	Microsoft Corporation
WmiPrvSE.exe		3,000 K	6,180 K	3284		
WmiPrvSE.exe		1,692 K	4,676 K	3112		
svchost.exe	< 0.01	2,288 K	4,800 K	724	Host Process for Windows S...	Microsoft Corporation
svchost.exe		12,788 K	9,780 K	768	Host Process for Windows S...	Microsoft Corporation
audiodg.exe		14,996 K	13,884 K	2132		
svchost.exe	< 0.01	26,288 K	29,300 K	884	Host Process for Windows S...	Microsoft Corporation
dwm.exe	0.37	61,748 K	62,500 K	1932	Desktop Window Manager	Microsoft Corporation
svchost.exe	0.01	14,644 K	20,400 K	924	Host Process for Windows S...	Microsoft Corporation
svchost.exe		4,924 K	7,660 K	1080	Host Process for Windows S...	Microsoft Corporation
svchost.exe	< 0.01	8,356 K	8,596 K	1224	Host Process for Windows S...	Microsoft Corporation
spoolsv.exe		6,984 K	6,756 K	1316	Spooler SubSystem App	Microsoft Corporation
svchost.exe		9,128 K	6,968 K	1352	Host Process for Windows S...	Microsoft Corporation
vmtoolsd.exe	0.20	5,424 K	8,236 K	1496	VMware Tools Core Service	VMware, Inc.
TPAutoConnSvc.exe	0.05	1,796 K	3,680 K	1696	ThinPrint AutoConnect printe...	Contado AG
TPAutoConnect.exe	0.02	2,980 K	6,392 K	1880	ThinPrint AutoConnect comp...	Contado AG
svchost.exe		1,020 K	2,800 K	1744	Host Process for Windows S...	Microsoft Corporation
dlhstc.exe	0.01	2,944 K	4,496 K	1956	COM Surrogate	Microsoft Corporation
msdtc.exe		2,392 K	3,288 K	428	Microsoft Distributed Transa...	Microsoft Corporation
svchost.exe		968 K	3,484 K	1140	Host Process for Windows S...	Microsoft Corporation
spssvc.exe		1,976 K	5,752 K	1760	Microsoft Software Protectio...	Microsoft Corporation
svchost.exe		140,080 K	27,072 K	684	Host Process for Windows S...	Microsoft Corporation
SearchIndexer.exe	0.01	15,360 K	8,004 K	1132	Microsoft Windows Search I...	Microsoft Corporation
taskhost.exe		6,880 K	6,528 K	1248	Host Process for Windows T...	Microsoft Corporation
lsass.exe		2,540 K	5,328 K	508	Local Security Authority Proc...	Microsoft Corporation
lsass.exe	0.02	1,232 K	2,540 K	516		
lsass.exe	0.66	4,496 K	11,028 K	424		
conhost.exe		636 K	2,248 K	964	Console Window Host	Microsoft Corporation
winlogon.exe		1,556 K	3,556 K	472		
explorer.exe	0.17	33,792 K	53,716 K	316	Windows Explorer	Microsoft Corporation
vmtoolsd.exe	0.36	21,452 K	25,604 K	1872	VMware Tools Core Service	VMware, Inc.
Tcpview.exe	0.70	5,296 K	11,340 K	3672		
procexp.exe	3.36	7,968 K	18,596 K	1876	Sysinternals Process Explorer	Sysinternals - www.sysinter...
firefox.exe		102,176 K	111,444 K	2804		

Firefox.exe and Tcpview.exe do not have a company name description. Malware authors frequently forget to add them when constructing malware.

## E. SCENARIO 5—TASK SCHEDULER ANALYSIS

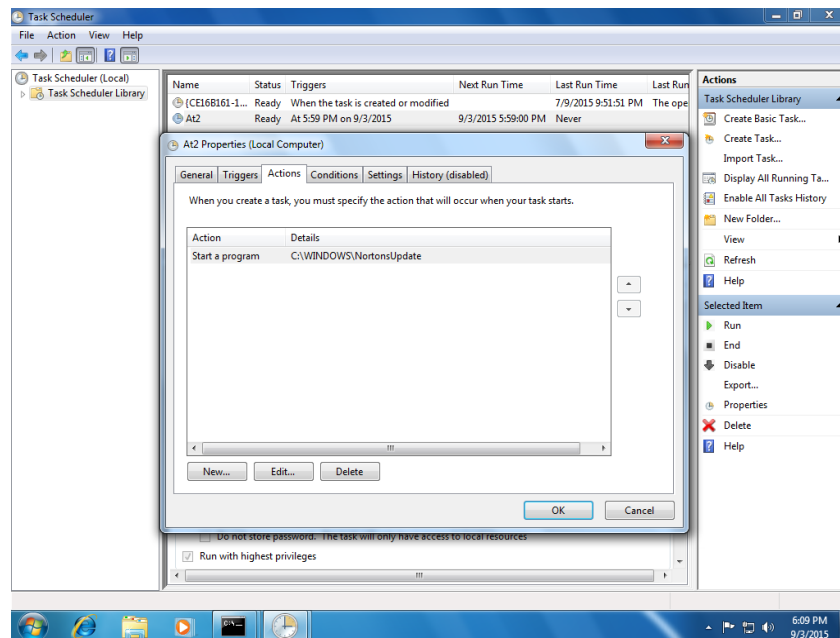
- (1) **Target Time to Complete:** 30 minutes
- (2) **Lab Description:** Investigate places within a computer that maybe used to automatically execute programs at a specific date and time and where these executables are stored. Screen shot suspicious tasks and/or locations where the binaries are stored for the lab write up.
- (3) **Expected Knowledge, Skills and Abilities to Complete:** The student must know where the Scheduled Task Manager can be found and that it can be used to execute malware. Also, only an administrator can schedule a task. The administrator account for this machine is Matthew.
- (4) **Source VM and System Requirements to Run:** In order to conduct this lab VMware Workstation must be installed. This lab scenario uses the Windows 7 operating system
- (5) **Lab Setup:** Malware will be downloaded into the Windows Task Scheduler. The malware is downloaded from the EC-Council CEHV8 Module 07 DVD located in the Viruses and Worms folder. The malware must be downloaded into the scheduler with an account other than

Matthew. Hundreds of good scheduled tasks will be placed in the scheduler to provide complexity for the lab. After the instructor places the good tasks and malware into the scheduler the instructor will then clone the VMware Workstation lab environment and name it Lab 5 for future student use.

## (6) Lab Write Up and Analysis

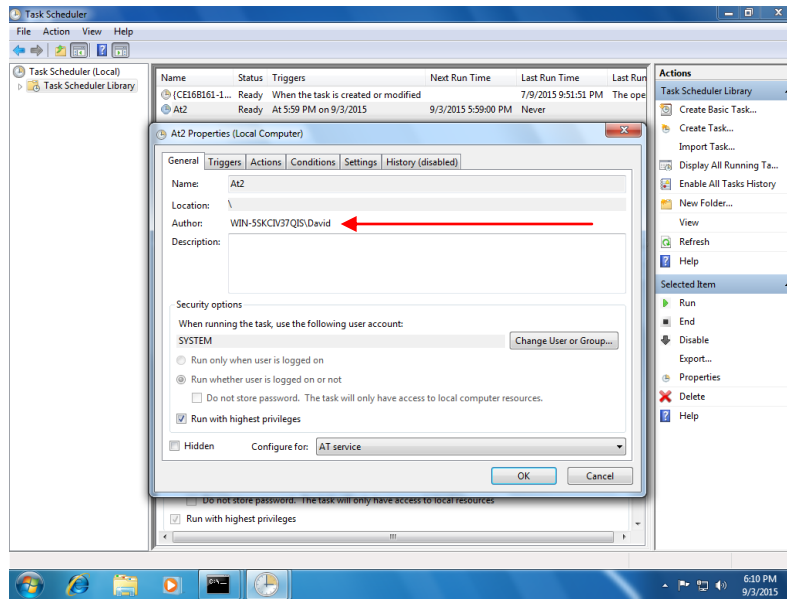
- (i) **Title:** Task Scheduler Analysis
- (ii) **Introduction:** The purpose of the lab is to teach the student how to inspect the Task Scheduler and spot malicious scheduled tasks. Sometimes it can be very difficult to determine if a task is malicious or not because many attackers are good at hiding in plain sight.
- (iii) **Tools:** “The Windows Task Scheduler provides the ability to automatically execute programs at a specific date and time or on a recurring basis” (Luttgens et al., 2014, p.305). This can be seen in Figures 16 and 17.
- (iv) **Results:** Provide screen shots with a brief description that convey pertinent indicators of compromise.

Figure 16. NortonsUpdate



This is a common place where attackers place malicious items if they only have command shell access.

Figure 17. Author's Username David



David is not the authorized administrator account.

## F. SCENARIO 6—EXECUTABLE ANALYSIS

- (1) **Target Time to Complete:** 45 minutes
- (2) **Lab Description:** Use PView to investigate 200 executables to determine if they contain malware. Suspicious events are captured using screen shots for the lab write up.
- (3) **Expected Knowledge, Skills and Abilities to Complete:** The student must know that malware utilizes thread local storage callbacks, which is used by malware to execute its code. The student must also know that the TLS table can be found in the IMAGE\_OPTIONAL\_HEADER, which is found under the IMAGE\_NT\_HEADERS using the PView tool. TLS callbacks are usually not used by applications that do not contain malware (Sikorski & Honig, 2012)
- (4) **Source VM and System Requirements to Run:** In order to conduct this lab VMware Workstation must be installed. This lab scenario uses the Windows 7 operating system.
- (5) **Lab Setup:** A malware executable will be placed into a folder on the desktop with 199 other non-malicious executables. The malware is downloaded from the viruses and worms folder on the EC-Council CEHV8 Module 07 DVD. Once the instructor adds the 199 non malicious executables and the executable with malware the instructor will then clone

the VMware Workstation lab environment and name it Lab 6 for future student use.

(6) **Lab Write Up and Analysis**

(i) **Title:** Executable Analysis

(ii) **Introduction:** The purpose of the lab is to teach the student how to operate and determine if an executable is malicious using PView. Sometimes it can be very difficult to determine if an executable is malicious or not, because many attackers are good at hiding malicious executables in plain sight

(iii) **Tools:** The Portable Executable (PE) View is a tool for viewing the PE file structure. It allows the user to view header information, individual sections, as well as import and export tables of functions operating on the system. Through the use of PView, thread local storage (TLS) callbacks can be effectively utilized (Sikorski & Honig, 2012). The executable is displayed in Figure 18 and its referencing the TLS callback is shown in Figure 19.

(iv) **Results:** Provide screen shots with a brief description that convey pertinent indicators of compromise.

Figure 18. View of the “Blem” Executable

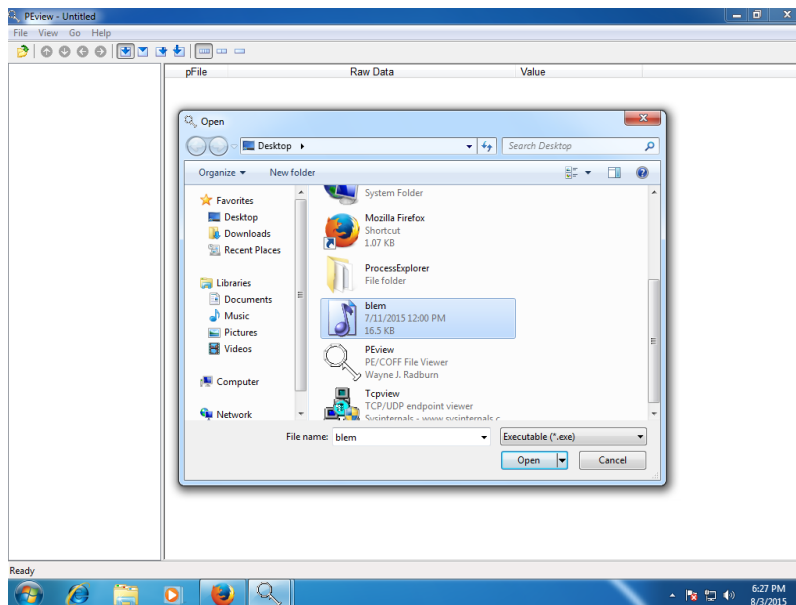
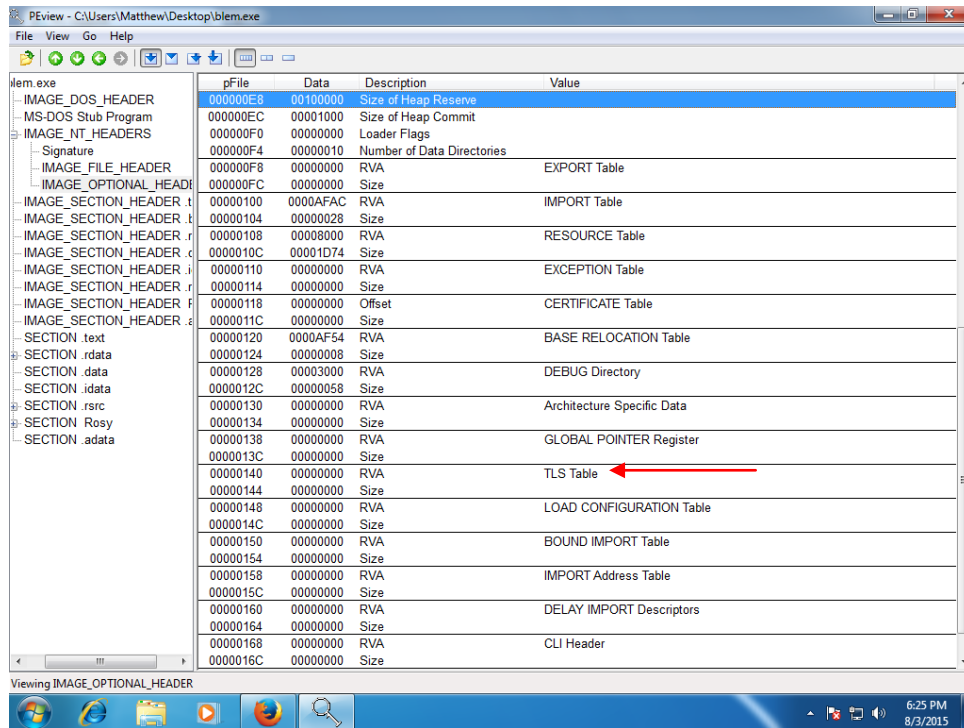




Figure 19. Blem.exe with a TLS Table in the Image Header



pFile	Data	Description	Value
000000E8	00100000	Size of Heap Reserve	
000000EC	00001000	Size of Heap Commit	
000000F0	00000000	Loader Flags	
000000F4	00000010	Number of Data Directories	
000000F8	00000000	RVA	EXPORT Table
000000FC	00000000	Size	
00000100	0000AFAC	RVA	IMPORT Table
00000104	00000028	Size	
00000108	00008000	RVA	RESOURCE Table
0000010C	00001D74	Size	
00000110	00000000	RVA	EXCEPTION Table
00000114	00000000	Size	
00000118	00000000	Offset	CERTIFICATE Table
0000011C	00000000	Size	
00000120	0000AF54	RVA	BASE RELOCATION Table
00000124	00000008	Size	
00000128	00003000	RVA	DEBUG Directory
0000012C	00000058	Size	
00000130	00000000	RVA	Architecture Specific Data
00000134	00000000	Size	
00000138	00000000	RVA	GLOBAL POINTER Register
0000013C	00000000	Size	
00000140	00000000	RVA	TLS Table
00000144	00000000	Size	
00000148	00000000	RVA	LOAD CONFIGURATION Table
0000014C	00000000	Size	
00000150	00000000	RVA	BOUND IMPORT Table
00000154	00000000	Size	
00000158	00000000	RVA	IMPORT Address Table
0000015C	00000000	Size	
00000160	00000000	RVA	DELAY IMPORT Descriptors
00000164	00000000	Size	
00000168	00000000	RVA	CLI Header
0000016C	00000000	Size	

The TLS Table indicates that blem.exe is potentially malware.

## G. SCENARIO 7—REGISTRY ANALYSIS

- (1) **Target Time to Complete:** 60 minutes
- (2) **Lab Description:** Use Regshot to complete a daily scan of the Registry. Screen shot anything suspicious for the lab write up.
- (3) **Expected Knowledge, Skills and Abilities to Complete:** The student will need to know the impossibility of going through the Registry line item by line item and how to use the Regshot tool. A Regshot image is taken every day at a specific time and is compared to the previous day's image to check for potential malicious content. The student must know how to spot potential malicious items on the output comparing the two snapshots.
- (4) **Source VM and System Requirements to Run:** In order to conduct this lab VMware Workstation must be installed. This lab scenario uses the Windows 7 operating system.
- (5) **Lab Setup:** Malware will be executed in the virtual environment. The malware is downloaded from the EC-council CEHV8 Module 07 DVD located in the Viruses and Worms folder. The instructor must take a snapshot of the Registry prior to the execution of the malware.

After the instructor takes the snapshot and executes the malware the instructor will then clone the VMware Workstation lab environment and name it Lab 7 for future student use.

(6) **Lab Write Up and Analysis**

(i) **Title:** Registry Analysis

(ii) **Introduction:** The purpose of the lab is to teach the student how to operate regshot and determine if anything malicious is hiding in the Registry. This is extremely difficult because the Registry has many files and it is constantly changing with each action conducted on the machine. In order to detect malicious items in the Registry network defenders must be constantly monitoring it with various tools.

(iii) **Tools:** Regshot allows for Registry comparison by taking snapshots of a pre-infected and post-infected system as seen in Figure 20. This allows the Incident Response team to identify changes made to the system by the malware as shown in Figure 21. Unfortunately, the results will often require plenty of patient scanning to decipher where the pertinent information is (displayed in Figure 22).

(iv) **Results:** Provide screen shots with a brief description that convey pertinent indicators of compromise.

Figure 20. First Registry Snapshot

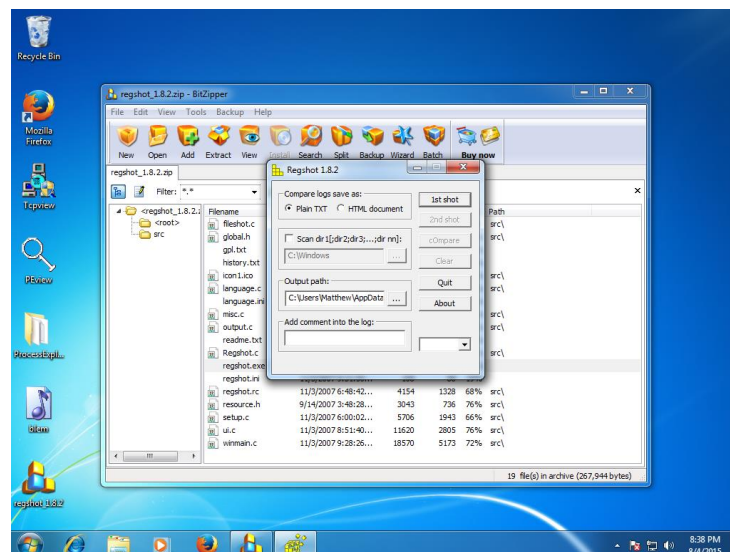


Figure 21. Second Registry Snapshot

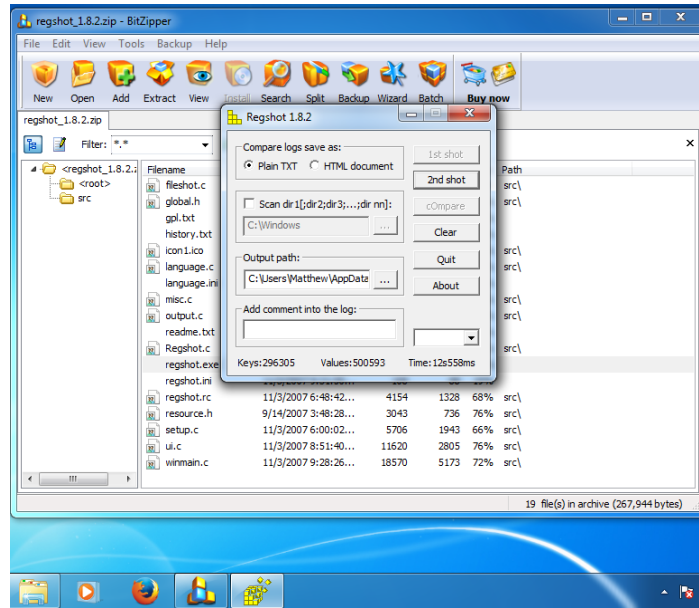
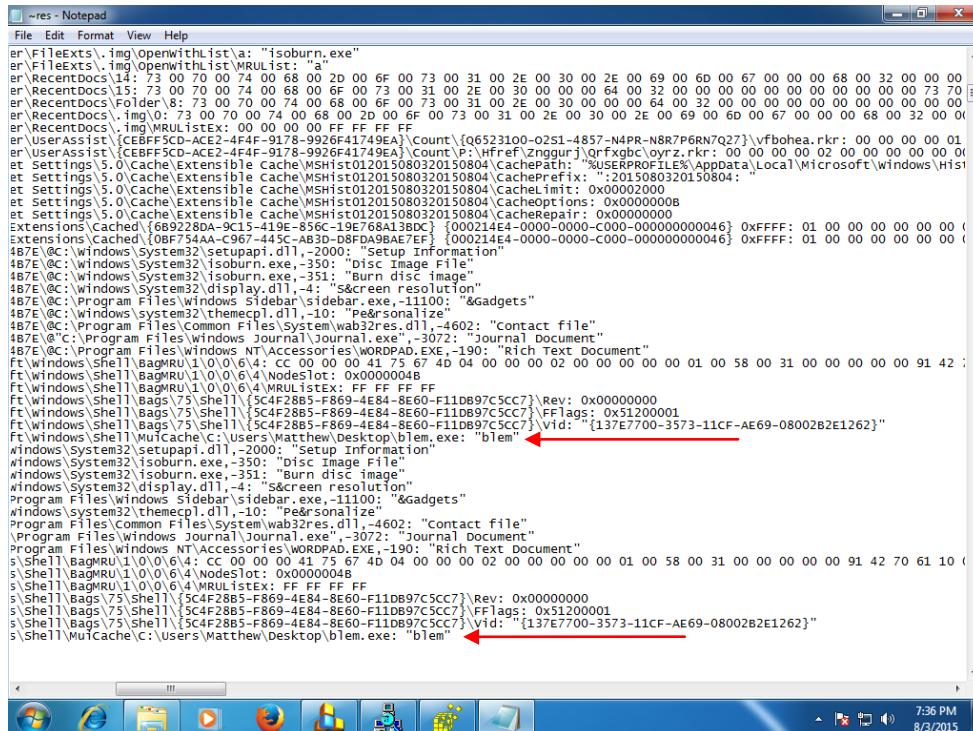


Figure 22. MUICache Display



MUICache signifies possible malware. When the program is investigated that generated the MUICache key more often than not, it can be identified as malware (Carvey, 2014).

THIS PAGE INTENTIONALLY LEFT BLANK

## **VI. CONCLUSIONS AND FUTURE WORK**

Cyber incident response incorporates all knowledge, skills, techniques, and tools used to prevent and detect malicious activity, as well as recover a corrupted system. The information provided here promotes the use of the virtual machines in incident response education. While incident response education can be delivered via lectures and reading, a way to imbue the incident responder with greater familiarity and confidence in the material is to provide realistic scenarios for training purposes. The best way to accomplish this is through the use of VMs.

### **A. CONCLUSIONS**

Incident response is an iterative process. This is to say the process entails a continuous cycling effort to: establish defenses, detect, identify and fix problems, and then learn from the experience in order to improve the process prior to the next incident. The technical terms used to explain these phases are Preparation, Identification, Containment, Eradication, Recovery, and Lessons Learned. The U.S. Navy, like other organizations, is highly dependent upon information systems and networks to conduct its various missions. Thus, it is important to ensure better preparation for incident handling, by improving the education of Navy personnel most directly relied upon to provide this vital service.

A way to learn how to effectively defend a system is through the use of exercises hosted on virtual machines (VMs). Chapter II covered the concept of VM and the many benefits of using them in incident response education. VMs can be used to provide students with hands-on experience with the types of attacks perpetrators use, as well as a way of directly interacting with and investigating them. A VM is self-contained and portable, which means it can be delivered via a drop box or flash drive without risk of harming the actual machine it is ultimately to be run (“played”) on. Should a student inadvertently make a mistake on their VM, they can revert (or “renew”) it back to its state prior to the mistake so long as snap shots have been taken throughout the activity conducted. This affords a more risk-free learning environment in which to practice and learn. VMs, thus, are convenient, low-risk as well as cost-effective for incident response training. They also

provide a convenient means with which to capture intentionally pre-infected systems that can then be used for incident response education.

Another aspect of good incident response training is the ability to isolate prominent IOCs (indicators of compromise). Chapter III enumerated several prominent system artifacts that would serve as logical starting points when performing a live incident response. These include the events, and various residual data that would (should) indicate to the student that an incident is likely to have occurred. Because the majority of Naval systems use the Windows Operating System (WOS), it was the system chosen for the exercises described in this study. The WinOS has many components that can potentially capture/indicate/reveal IOCs. The Registry, processes, certain files, network connections, tasks, accounts, logs, and memory and disk information all provide system artifacts that can be investigated for signs of infection/exploitation. Thorough knowledge of these aspects of the WinOS lends insight into what is and is not functioning normally, as well as pointing the investigation to other likely indicators and evidence.

In addition to the operating system, network services and applications are areas that can also be mined for potential IOCs. Commonly targeted services are DHCP, DNS, Web, and email. These are often taken advantage of by hackers through many different tactics. It is equally important that an incident responder know the intricacies of these items and where the “weaknesses” lie, as is having a broad understanding of the host operating system that they run on.

There are numerous investigative tools that can be utilized in support of an incident investigation. Rather than attempt to cover all of them, this study focused on those that are most prominent (i.e., frequently used) in the area of incident response. These consist of dedicated add-on utilities, as well as OS-native command line programs. Some dedicated utilities often used are the Quick Checksum Verifier, PView, Process Explorer, TCPView, and Regshot. Some common OS-native command line tools and capabilities include *netstat* event viewer, viewing running processes and services, and verifying system file integrity checksums. All these contribute to additional system IOC examination.

Incident response differs from digital forensics in that an incident responder can be likened to a medical first responder (paramedic), whereas a forensics expert would be

comparable to a surgeon. An incident responder is intended to provide quick and immediate assistance to get a system back to proper functioning order (breathing restored and bleeding stopped by paramedic); whereas a forensics expert has the more sophisticated knowledge and tools needed to dig deeper into the root cause(s) of an incident (treatment of occluded artery by surgeon). Because of the heavy reliance on technology by the Navy, it is imperative that a robust incident response capability be maintained. The use of virtual machines in this capacity can prove very valuable. Virtual machines provide a “risk-free” environment for students to manipulate and interact with key investigative tools in order to identify, scope, contain and perhaps eliminate system exploits.

In furtherance of the above articulated goal of advancing incident response teaching via pre-infected VMs, seven compromised system “scenarios” were created that entailed analysis of the principal first responder WinOS artifacts (PUFN TAL); each captured in a separate VM. Through generating these scenarios and the research conducted here, we have made informed decisions regarding *which* of the many potential “attack craft” artifacts should be represented in this set of compromised systems. The scenarios presented here provide students with the opportunity to apply their knowledge of incident response tools, IOCs, and methodologies, as they go about using tools, knowledge and methodology to identify the IOCs in each scenario.

## **B. FUTURE WORK**

The idea described in this study was fairly narrow in scope: identify the dominant WinOS artifacts used in incident response investigation, capture them in separate scenario-based pre-infected VMs, then discuss the tools useful in conducting such investigations. This scope could be greatly widened by developing additional incident response VM scenarios. Likely candidates for additional scenarios include: a) examining different OSs, b) adding additional artifact types (e.g., network traffic and firewall logs), c) creating *hybrid* incident scenarios that require the investigator to correlate several artifact types, and d) adding additional analysis tools.

Gone are the days of Windows reigning as the champion (and pretty much sole) OS used by the masses. While the majority still utilize Windows and various versions of it, today there are more options. Ubuntu, Linux Mint, Macintosh OSX, Android and Fedora

are all alternatives to the dominant Windows OS. Additional research can be conducted as related to the information provided here to this end. Separate, dedicated, VMs can be set up that address these other alternatives, allowing the individual to extend their knowledge of incident response as it pertains to an OS other than Windows. In addition, because many servers are UNIX/Linux based they are often a target for malicious attacks. The research could be broadened to take this into account

More artifact types can be examined. The Windows Registry keys and values and DNS queries were discussed here. However, others such as the persistence mechanism—which allows malware to survive reboots and logins—specifically could be mined for future work. In addition, artifacts from Skype, Facebook chat, and other instant messaging services may be further researched. These items are specifically interesting because they are commonly used by Naval personnel for communication purposes as they are frequently away from their families.

Only a few types of incidents have been delved into here. The idea was to focus on those that are most common so as to concentrate the type of education provided. This being stated, with time comes evolution of attackers' methods. Because of this, the kinds of incidents to experiment with are many. An extension of the work submitted here would be to provide alternate incidents for examination.

There is much that can be done to expand upon the ideas addressed here. The use of VM in incident response promises to be extremely beneficial to students for training purposes and testing the effectiveness of these scenarios will exemplify this. It not only provides the opportunity to make the best use of their knowledge in a hands-on yet controlled environment; but it also allows for experimentation with multiple types and severities of incidents. The implication of the use of VMs as a source of education needs to be established as criteria for graduate work as well as incident response training throughout the Navy.



## LIST OF REFERENCES

- Andress, J., & Winterfeld, S. (2014). *Cyber warfare techniques, tactics and tools for security practitioners*. Waltham, MA: Elsevier.
- Arquilla, J. (2011). From blitzkrieg to bitskrieg: The military encounter with computers. *Communications of the ACM*, 54(10), 58–65. doi: 10.1145\*2001269.2001287
- Carvey, H. (2014). *Windows forensic analysis toolkit* (4th ed.). Waltham, MA: Elsevier.
- Daryabar, F., Dehghantanha, A., & Broujerdi, H. G. (2011). Investigation of malware defense and detection techniques. *International Journal of Digital Information and Wireless Communications*, 1(3), 645–650.
- Dezfoli, F. N., Dehghantanha, A., Mahmoud, R., Nor Fazlida Binti Mohd Sani, & Daryabar, F. (2015). Digital forensic trends and future. *International Journal of Cyber-Security and Digital Forensics*, 2(2), 48–76.
- Eom, J., Kim, N., Kim, S., & Chung, T. (2012). Cyber military strategy for cyberspace superiority in cyber warfare. *Cyber Warfare and Digital Forensic 2012 International Conference on Cyber Security, USA*, 295–299. doi 10.1109/CyberSec.2012.6246114
- Forouzan, B. (2010). *TCP/IP protocol suite* (4th ed.). New York: McGraw Hill.
- Geers, K. (2012). Strategic cyber defense: Which way forward? *Journal of Homeland Security & Emergency Management*, 9(1), 1–10. doi:10.1515/1547-7355.1868
- Gervais, M. (2012). Cyber attacks and the laws of war. *Berkeley Journal of International Law*, 30(2), 525–579.
- Golden, B. (2008). *Virtualization for dummies*. Hoboken, NJ: Wiley Publishing.
- Guinchard, A. (2011). Between hype and understatement: reassessing cyber risks as a security strategy. *Journal of Strategic Security*, 4(2), 75–95. doi:10.5038/1944-0472.4.2.5
- Hui, Z., & Yanwei, S. (2012). Research on network attack-defense training based on virtual monitor. *Journal of Convergence Information Technology*, 7(21), 228–235. doi:10.4156/jcit.vol7.issue21.29
- Luttgens, J., Pepe, M., & Mandia, K. (2014). *Incident response & computer science* (3rd ed.). New York: McGraw-Hill Education.

- Naval Postgraduate School (2013, April 1) *Cyber Academic Group, GSOIS CAG*. Retrieved from <https://www.nps.edu/Academics/Schools/GSOIS/Departments/CAG/Facilities/facilities.html>
- Price, M. (2008). The paradox of security in virtual environments. *Computer*, 41(11), 22–28. doi:10.1109/MC.2008.472
- Regalado, D., Harris, S., & Harper, A. (2015). *Gray hat hacking: The ethical hacker's handbook* (4th ed.). New York: McGraw-Hill Education.
- Schiffman, M., O'Donnell, A. J., Pennington, B., & Pollino, D. (2003). *Hacker's challenge 2 test your network security & forensic skills*. Berkeley, CA: McGraw-Hill/Osborne.
- Sikorski, M., & Honig, A. (2012). *Practical malware analysis*. San Francisco: No Starch Press.
- Skoudis, E., & Liston, T. (2006). *Counter hack reloaded: A step-by-step guide to computer attacks and effective defenses* (2nd ed.). Boston: Pearson Education.
- Slocombe, G. (2012). Cyber security Web War II. *Asia-Pacific Defence Reporter*, 38(4), 38–40.
- Terry, C., Castellano, A., Harrod, J., Luke, J., & Reichherzer, T. (2014, January). The UWF cyber battle lab: A hands-on computer lab for teaching and research in cyber security. In *Proceedings of the International Conference on Security and Management (SAM)* (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).
- Tondel, I., Line, M., & Jaatun, M. (2014). Information security incident management: Current practice as reported in the literature. *Computers & Security*, (45), 42–57. Retrieved from [www.elsevier.com/locate/cose](http://www.elsevier.com/locate/cose)
- Van Dusen, M. (2013). *Cyber-warfare—a legitimate concern?*. Unpublished manuscript.
- Wilson, J. (2014). Cyber warfare ushers in 5th dimension of human conflict. *Military & Aerospace Electronics*, 25(12), 8–15.

## **INITIAL DISTRIBUTION LIST**

1. Defense Technical Information Center  
Ft. Belvoir, Virginia
2. Dudley Knox Library  
Naval Postgraduate School  
Monterey, California